Doctor of Philosophy

Engineering

Institute for Risk and Uncertainty

Robust and Efficient Probabilistic Approaches towards Parameter Identification and Model Updating

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor of Philosophy

by:

Adolphus Lye Tee Siang
B.Sc (Hons), Physics

Supervised by:

Professor Edoardo Patelli (Primary)
Professor Alice Cicirello (Secondary)

May 22, 2023

Dedication

This dissertation represents 4 years of blood, sweat and (plenty of) tears that was put into the PhD study and the journey.

I dedicate this to my teachers past and present who I am so fortunate to meet through the course of my formal education journey. Without their teachings and encouragements, I would not have made it thus far in my studies.

To my peers who I am so glad to have met from back home (in Singapore) and here (in the UK), thank you for making my life much more vibrant and my PhD journey less miserable.

To my colleagues within the Institute for Risk and Uncertainty, thank you for all your help, advice, and the moral support over the years which have gone a long way in providing more certainty as to what I need to do through the course of my PhD journey.

To the University of Liverpool Athletics and Cross Country Team, thank you for always being there for me and for providing me with many opportunities to relive the Undergraduate life that I never initially had through the socials and the unique team-spirit. Without which, life would have been so much more boring.

To my fellow Residential Advisor colleagues in Vine Court, my hall of residence for the past 4 years, thank you for your friendship and support you have rendered me and making this hall a home away from home.

To the fellows of St. Philemon's Church, thank you so much for all the kindness you have rendered me along the way and helping me settle into the Liverpool community since I first arrived. I am truly grateful for your wonderful gestures and prayers you have given me throughout. It will never be forgotten.

To those of us who have at one point of time been told that they are not good enough or have road-blocks put in their way of achieving their dreams, never give up and continue the fight. There is a classic saying: "Fight for your dreams, and your dreams will fight for you." - Bryan Danielson (2019)

To my beloved family members in Singapore, thank you so much for your constant support and trust in me from half-way across the world which has never been more valuable without which I would not be able to sustain throughout the journey.

And most importantly, to God whose unwavering presence helped provide a sense of assurance in one way or another especially in the uncertain times of the COVID-19 pandemic and the many times when the PhD going seemed impossible.

Thank you to all. Trust the Process.

Acknowledgements

The author would like to express his thankfulness and gratefulness to the Singapore Nuclear Research and Safety Initiatives and the National Research Foundation for their gracious funding, without which, this PhD study and the study abroad experience with the University of Liverpool would not be possible.

The contributions of Dr. Roberto Rocchetta and Dr. Matteo Broggi in Chapter 5 is gratefully acknowledged. Both Dr. Roberto Rocchetta and Dr. Matteo Broggi provided the data and the Artificial Neural Network surrogate model required to implement the Bayesian model updating approach on the shear aluminium frame which was also studied by them in [1].

The contributions of Dr. Masaru Kitahara and Dr. Matteo Broggi in Chapter 6 is immensely acknowledged. Both Mr. Masaru Kitahara and Dr. Matteo Broggi provided the tools and the distribution-free approach towards setting up an Uncertainty Model to address the problem presented in the recent NASA-Langley Uncertainty Quantification Challenge 2019. Without their inputs and the time they have given me to address my queries, I would not have gained further conceptual understanding towards the numerous tools and approaches within the domain of Uncertainty Quantification.

The contributions of Dr. Luca Marino in Chapter 8 is tremendously acknowledged. Dr. Luca Marino assisted not only in taking me through the physics behind the Coulomb damping of the structure being studied, but also took time from his PhD studies to run the experiments and obtain the necessary data for the required analysis of the work. Without his help and guidance, this work would not have come into fruition and I would not have had gained further understanding behind the significance of Coulomb friction.

The author gratefully acknowledges the contributions by Professor Nawal Prinja (Prinja and Partners) towards the feasibility study presented in Chapter 9. Professor Nawal Prinja helped to provide the experiment data from industrial data-base as well as the Artificial Neural Network models for the study. With his expertise in the field stemming from 4

decades of academic and industrial experience in the nuclear sector, as well as his kindness and patience despite my relative lack of knowledge in that field, it has served as a great inspiration and making the collaboration experience a pleasant one.

The author would also like to express his gratitude to the Internal and External examiners for their valuable time in proof-reading and providing detailed, constructive, and insightful suggestions towards improving the quality of the dissertation.

Penultimately, the author would like to express his immense appreciation and thankfulness to Professor Scott Ferson, the Chair in Risk and Uncertainty and the Director of the Institute for Risk and Uncertainty, for not only the years of fruitful discussions and teachings which helped improved my understanding on the subject of Uncertainty quantification, but also for the moral support and empathy he rendered to the fellow PhD students and myself whenever we face any problems in the course of our studies. I wish him well and the continued success in his research and teaching endeavours.

Finally, the author is deeply grateful to his supervisory team, consisting of Professors Edoardo Patelli and Alice Cicirello, for their co-supervison. While the journey was never smooth-sailing, especially in the midst of the COVID-19 pandemic, I am thankful for the time they have set aside for me over the last 4 years to address my doubts, advice, motivate and guide me through the process. I wish them both well and the continued success in their respective careers ahead.

Abstract

In engineering, the virtual behaviour of structures under operational and extreme conditions are investigated using mathematical or physics-based models. To obtain numerical responses that best reflect the structure under investigation, the physical input parameters describing the geometric, material, and damping properties of these models need to be identified or inferred.

However, the presence of uncertainty poses significant challenges in parameter identification. Often, these uncertainties would stem from the following: 1) the aleatory uncertainty due the variations in the response measurements of nominal identical structures under same loading conditions due to manufacturing and material variability, thus, leading to the parameter not having a single "true" parameter value representation; 2) the epistemic uncertainty associated with the "fuzziness" to the knowledge of the parameter(s) as a result of the experimental data/measurements being usually affected by "noise"; and 3) the model uncertainty due to the modelling errors associated with the failure of the model in capturing the physics of the problem. This presents the need to not only perform an inference on the parameter(s), but also quantify the uncertainty associated with the estimates. An approach towards this would be Bayesian model updating, which serves as the context of the dissertation.

The dissertation provides details to the efficient and robust approaches towards probabilistic parameter identification and model updating via the aforementioned approach. To realize this, an extensive literature review on Bayesian inference and the existing sampling tools is provided. This is done to identify the key research gaps, as well as limitations to the current sampling algorithms. From there, the Transitional Ensemble Markov Chain Monte Carlo sampler is proposed to which its strengths include its robustness in sampling from skewed distributions, quicker computational time, and the removal of any need for tuning by the users. To demonstrate this, the algorithm has been implemented on both numerical and real-world examples. The latter involves a structural health monitoring problem and the recent NASA-Langley Uncertainty Quantification challenge. Following which, the analysis is extended towards inferring time-varying parameter(s) via on-line Bayesian inference. This motivated the development of the Sequential Ensemble Monte Carlo sampler to which its strengths include its robustness in identifying the most probable Markov kernel under uncertainty. Such strengths are demonstrated through the experimental example involving a single-storey structure subjected to a time-varying Coulomb friction. Finally, the dissertation presents an approach to merge Artificial Intelligence tools with Bayesian statistics towards the probabilistic prediction of material properties for Nuclear power plant structures. Such development seeks to enable the Artificial Intelligence models to provide a more robust probabilistic prediction on the material properties under very limited data and model uncertainty.

For the interest of the relevant practitioners, the algorithms to the proposed methods presented in the dissertation are made accessible on OpenCOSSAN, an open-source software for uncertainty quantification, as well as GitHub.

Declaration

I declare that the dissertation was composed by myself and that the results contained herein have not been submitted for any other degree or professional qualification. I confirm that the work submitted is mine, except where it forms part of a jointly-authored publication. In which case, my contribution and the names of the other authors have been explicitly indicated below. I also confirm that appropriate credit is given where reference is made to others' work and that the dissertation contains: 230 pages, 88 figures, 58 tables, and 43697 words.

Pursuant to the contents presented in this dissertation, I declare that:

- Chapters 2 and 3 of this dissertation are based on the peer-reviewed literature titled: "Sampling methods for solving Bayesian model updating problems: A tutorial", co-authored by A. Cicirello (my secondary supervisor) and E. Patelli (my primary supervisor), published as part of the Special Issue on Advances in Stochastic Model Updating within the Journal of Mechanical Systems and Signal Processing;

- Chapters 4 and 5 of this dissertation are based on the peer-reviewed literature titled: "An efficient and robust sampler for Bayesian inference: Transitional Ensemble Markov Chain Monte Carlo", co-authored by A. Cicirello (my secondary supervisor) and E. Patelli (my primary supervisor), published in the Journal of Mechanical Systems and Signal Processing;

- Chapter 6 of this dissertation is based on the peer-reviewed literature titled: "Robust optimization of a dynamic Black-box system under severe uncertainty: A distribution-free framework", co-authored by M. Kitahara, M. Broggi and E. Patelli (my primary supervisor), published as part of the Special Issue on The NASA Langley Challenge on Optimization under uncertainty within the Journal of Mechanical Systems and Signal Processing;

- Chapters 7 and 8 of this dissertation are based on the literature titled: “Sequential Ensemble Monte Carlo sampler for On-line Bayesian inference of Time-varying Model parameters in Engineering Applications”, co-authored by L. Marino, A. Cicirello (my secondary supervisor) and E. Patelli (my primary supervisor), published in the ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering; and
- Chapter 9 of this dissertation is based on the peer-reviewed literature titled: “Probabilistic Artificial Intelligence Prediction of Material Properties for Nuclear Reactor Designs”, co-authored by N. Prinja and E. Patelli (my primary supervisor), published in the Proceedings of the 32^{st} European Safety and Reliability Conference 2022.

Adolphus Lye Tee Siang
May 22, 2023

List of Publications

Peer-reviewed Journal Publications

1. Adolphus Lye, Alice Cicirello, and Edoardo Patelli. Sampling methods for solving Bayesian model updating problems: A tutorial. *Mechanical Systems and Signal Processing,* ***159***, 107760, 2021. doi: `10.1016/j.ymssp.2021.107760`

2. Adolphus Lye, Alice Cicirello, and Edoardo Patelli. An efficient and robust sampler for Bayesian inference: Transitional Ensemble Markov Chain Monte Carlo. *Mechanical Systems and Signal Processing,* ***167***, 108471, 2022. doi: `10.1016/j.ymssp.2021.108471`

3. Adolphus Lye, Masaru Kitahara, Matteo Broggi, and Edoardo Patelli. Robust optimisation of a dynamic Black-box system under severe uncertainty: A Distribution-free framework. *Mechanical Systems and Signal Processing,* ***167***, 108522, 2022. doi: `10.1016/j.ymssp.2021.108522`

4. Adolphus Lye, Alice Cicirello, and Edoardo Patelli. Sequential Ensemble Monte Carlo sampler for On-line Bayesian inference of Time-varying Model parameters in Engineering Applications. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, 1-13, 2023. doi: `10.1115/1.4056934`

5. Adolphus Lye, Alice Cicirello, and Edoardo Patelli. An investigation towards the Uncertainty Model calibration approaches for NASA-Langley UQ Challenge 2019. *Journal of Physics: Conference Series*, 2023. (Accepted)

Peer-reviewed Conference Publications

1. Adolphus Lye, Hector Diego Estrada-Lugo, and Edoardo Patelli. Conversion of Fault Tree into Credal Network for Probabilistic Safety Assessment of a Nuclear Power Plant. *In the Proceedings of the* 3rd *International Conference on Nuclear Power Plants: Structures, Risk and Decommissioning*, London, 2019.

2. Adolphus Lye, Alice Cicirello, and Edoardo Patelli. Uncertainty Quantification of Optimal Threshold Failure Probability for Predictive Maintenance using Confidence Structures. *In the Proceedings of the* 3rd *International Conference on Uncertainty Quantification in Computational Sciences and Engineering*, Crete, 2019. doi: 10.7712/120219.6364.18502

3. Adolphus Lye, Alice Cicirello, and Edoardo Patelli. A Review of Stochastic Sampling Methods for Bayesian Inference Problems. *In the Proceedings of the* 29th *European Safety and Reliability Conference*, Hannover, 2019. doi: 10.3850/978-981-11-2724-3_1087-cd

4. Adolphus Lye, Alice Cicirello, and Edoardo Patelli. Bayesian Model Updating of Reliability Parameters using Transitional Markov Chain Monte Carlo with Slice Sampling. *In the Proceedings of the* 30th *European Safety and Reliability Conference*, Venice, 2020. doi: 10.3850/978-981-14-8593-0_4374-cd

5. Adolphus Lye, Ander Gray, and Edoardo Patelli. Identification of Time-varying Parameters using Variational Bayes - Sequential Ensemble Monte Carlo Sampler. *In the Proceedings of the* 31st *European Safety and Reliability Conference*, Angers, 2021. doi: 10.3850/978-981-18-2016-8_081-cd

6. Adolphus Lye, Alice Cicirello, and Edoardo Patelli. On-line Bayesian Model Updating and Model Selection of a Piece-wise model for the Creep-growth rate prediction of a Nuclear component. *In the Proceedings of the* 8th *International Symposium on Reliability Engineering and Risk Management 2022*, Hannover, 2022. doi: 10.3850/978-981-18-5184-1_MS-02-208-cd

7. Adolphus Lye, Alice Cicirello, and Edoardo Patelli. On-line Bayesian Inference for Structural Health Monitoring under Model Uncertainty using Sequential Ensemble Monte Carlo. *In the Proceedings of the* 13th *International Conference on Structural Safety and Reliability*, Shanghai, 2022. (Accepted)

8. Adolphus Lye, Nawal Prinja, and Edoardo Patelli. Probabilistic Artificial Intelligence Prediction of Material Properties for Nuclear Reactor Designs. *In the Proceedings of the* 32^{st} *European Safety and Reliability Conference*, Dublin, 2022. doi: `10.3850/978-981-18-5183-4_S24-02-306-cd`

9. Adolphus Lye, Nawal Prinja, and Edoardo Patelli. Towards a Robust Prediction of Material properties by Artificial Intelligence and Probabilistic methods. *In the Proceedings of the 7th International Conference on Topical Issues in Nuclear Installation Safety*, Vienna, 2022. (Accepted)

10. Adolphus Lye, Ander Gray, Marco de Angelis, and Scott Ferson. Robust Probability Bounds Analysis for Failure Analysis under Lack of Data and Model Uncertainty. *In the Proceedings of the* 5^{th} *International Conference on Uncertainty Quantification in Computational Sciences and Engineering*, Athens, 2023. (Accepted)

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Abstract

The chapter provides the context of the dissertation by first providing a background description on the concept of model updating and its importance within the engineering discipline. This includes an overview of the 2 distinct approaches to model updating: 1) the deterministic; and 2) the probabilistic approaches to which an evaluation to the pros and cons to each approach are provided. Following this, the research challenges are highlighted and the research objectives to be achieved in the dissertation are identified. Finally, the chapter concludes with a detailed explanation to the structure of the dissertation and its content organisation.

1.1 Background

In engineering design problems, mathematical models are used to investigate the virtual behaviour of structures under operational and extreme conditions. Generally, such models describe the assumed relationship between the input and output variables of such structures being studied. In order to obtain numerical responses representative of the structure under investigation, the physical input parameters describing the geometric, material and damping properties of these models need to be updated [10]. This is known as model updating and one such conventional example would be the structural Finite Element model updating [11, 12].

Under such approach, the input parameters of the physical model are updated or optimised through the process of minimising the difference between the experimental and modelling results with respect to a suitable response metric which is sensitive to the variation of such input parameters [13]. For example, the parameter(s) of a mathematical model describing the material properties of a plate can be updated in order to minimise the difference between the theoretical and experimental natural frequencies of the plate. Ideally, one seeks to achieve the case where the input model parameters are well optimised such that this discrepancy between the model output and experimental observations is close to 0.

Model updating in engineering has been implemented for numerous key reasons including [13]: 1) design verification and validation of structures; 2) to provide improved model predictions on the structural response quantities such as its frequency response functions; and 3) to identify the unknown system characteristics such as the extent of the localised structural damage in structural health monitoring. In regards to the third point, one way this is achieved would be through the monitoring of the structure's stiffness parameter(s). This is under the assumption that the localised structural damage leads to a local reduction in the stiffness. Hence, this highlights a benefit of the model updating approach, in that it allows for a non-destructive means to monitor thoroughly and accurately the health condition of the structure being studied [13].

However, the technique of model updating faces 3 main challenges: 1) it assumes that the mathematical model employed is able to capture the physics of the problem in full and is unaffected by modelling errors; 2) it does not readily take into account that the experimental data are usually affected by "noise" [14, 15, 16, 17]; and 3) it does not consider that response measurements of nominal identical structures under same loading conditions might vary because of manufacturing and material variability, which should be included in the model by considering input parameter variability [18, 19, 20, 21, 22] and not a single "true" parameter value representation [23].

The aforementioned challenges introduce elements of uncertainty which should also be accounted for in the model updating procedure. Such uncertainties can be classified into 2 distinct types [24, 25]: 1) Aleatory (i.e. Type A uncertainty); and 2) Epistemic (i.e. Type B uncertainty). The aleatory uncertainty refers to the statistical uncertainty of a given variable as a result of its inherent variability and randomness [26]. Such uncertainty is irreducible and are usually described or quantified with a probability distribution [27, 18]. On the other hand, the epistemic uncertainty refers to the lack of knowledge over the unobserved parameter(s) or variable(s). However, unlike the aleatory uncertainty, epistemic uncertainties are reducible with increased information or data through model updating [28]. An epistemic uncertainty over a parameter is generally represented by a fixed value within a bounded set whose intervals reflect the level of knowledge on the parameter [26]. The lower the level of knowledge, the larger the interval of this bounded set.

Broadly speaking, the approaches towards model updating under uncertainty can be grouped into 2 categories: 1) deterministic; and 2) probabilistic approach.

1.1.1 Deterministic approach

In the deterministic approach, the model updating procedure does not account for any form of uncertainty that is associated with the inferred parameter(s). Through this method, the calibration of the inferred parameter(s) of a single model is performed based on one set of test data [3]. This yields a single set of crisp values on the inferred parameter(s), giving rise to a single model prediction with maximum fidelity given the single set of data [29]. Examples of such approach towards model updating include: 1) the Linear least-squares minimization [30]; and 2) Sensitivity-based model updating [3].

A general advantage to the deterministic approaches is that they are relatively systematic in its implementation after decades of development [31]. However, its disadvantages include: 1) the issue of high computational costs when performing global optimisations on the inferred parameter(s); 2) it does not account for the uncertainty which can be attributed to the test data or the model for example; and 3) it only provides one single solution to the updated model predictions and the inferred parameter(s) and neglects other possible solutions with equal importance [31, 32].

The last point is worth significant attention given that this makes the deterministic approach insufficiently robust in the context of uncertainty quantification which requires the consideration of probabilistic predictions of the model and the probabilistic estimates of the inferred parameter(s). Hence, in such case, one can turn to and implement the probabilistic approach towards model updating.

1.1.2 Probabilistic approach

In the probabilistic approach, the model updating procedure considers not just solely the presence of epistemic uncertainty, but also the aleatory uncertainty. Such condition is referred to as “hybrid uncertainties” [33]. In such case, imprecise probability models are required to model the inferred parameter(s) whereby its associated uncertainty can be aleatory but the uncertainty over its distribution hyper-parameters (i.e. the mean and variance parameters) can be epistemic. Examples of such imprecise probability models include: 1) the Probability-box (P-box) [34]; 2) Evidence theory [35]; and 3) Fuzzy-probability model [36].

Unlike the deterministic approach, the probabilistic approach does not provide point-estimate(s) to the inferred parameter(s), but rather in the form of intervals or probability distributions. As such, this yields a set of probabilistic model predictions which are able to illustrate the uncertainty associated with the given observations [33].

Among the probabilistic approaches, one of the most well-established probabilistic approaches is the Bayesian model updating framework [37, 38]. This framework shall serve as the context of discussion and the basis of the research works presented in the dissertation. Through the Bayesian model updating approach, the physical parameters of the model to be updated are represented by probability distributions, and Bayesian inference is employed to evaluate the posterior probability density function given some measured data. In the structural health monitoring community, such form of statistical model updating is often referred to as system identification [39, 40, 41, 42, 43, 44, 45]. For the real case applications, probabilistic model updating relies on the availability of efficient sampling techniques. This is due to the relative complexity of the distribution from which samples are generated. Detailed discussions to this will be covered in Chapter 2 of the dissertation.

1.2 Research motivations

The Bayesian model updating approach mostly requires the implementation of Monte Carlo methods in order to address probabilistic model updating problems, quantify the uncertainty of the inferred parameter(s), and identify the most probable model under uncertainty and lack of data [46]. There are 2 key reasons for this: 1) the mathematical form of the target distribution (i.e. the posterior distribution) cannot be expressed in a closed-form in general; and 2) the posterior distribution is mostly not normalised. These factors present challenges in computing analytically the statistics of the estimates, such as their means and variances. To overcome such issues, numerical approaches are required to estimate these statistics in the form of Monte Carlo techniques to which examples include: 1) Multiple-try Metropolis Algorithm [47]; 2) Metropolis Adjusted Langevin Algorithm [48]; and 3) Hamiltonian Monte Carlo [49].

In the dissertation, the discussions will focus mainly on the following 3 Monte Carlo techniques: 1) Markov Chain Monte Carlo [50]; 2) Transitional Markov Chain Monte Carlo [51]; and 3) Sequential Monte Carlo [52] samplers. Details on their respective algorithm will be reviewed in Chapter 3 of the dissertation.

1.2.1 Research challenges

Over the years, there have been numerous research which look towards developing state-of-the-art Monte Carlo sampling approaches which are more robust and efficient in generating samples whose distribution is most representative of the corresponding posterior distribution for Bayesian inference. These recent developments are discussed in Chapters 3, 4, and 7 of the dissertation. While these developments have demonstrated improved sampling performance and provided improved inference results based on their respective literature, they still face the following challenges:

1) The ineffectiveness in sampling from highly-skewed (i.e. complex-shaped) distributions which can have a direct impact on how well the samples represent the distribution of the posterior and the resulting estimates of the inferred parameter(s);

2) The lack of robustness in moderating acceptance rates within optimal bounds which can affect the performance of the sampler(s) in exploring the sample space of the posterior;

3) The requirement of manual tuning, which can be a tedious process when dealing with problems where there is no information apriori to justify a choice of value(s) for the tuning parameter resulting in many rounds of such tuning procedure in a "trial-and-error" fashion;

4) The high computational costs incurred and reduced sampling efficiency when dealing with

complex problems involving the need to infer/estimate a high number of parameters (e.g. ≥ 18) [53];

5) The need to improve the robustness of the inference and model updating procedure by accounting for the model uncertainty; and

6) The difficulty in performing probabilistic model updating and prediction given sparse data and limited information over the system of interest.

1.2.2 Research objectives

The overall research aim is to develop robust and efficient sampling approaches towards parameter identification and model updating. This is achieved through a series of thorough research works presented in the dissertation which seek to address the challenges highlighted in Section 1.2.1. Hence, the research objectives to be accomplished in the dissertation are as follows:

1) To provide a detailed comparison and evaluation between the 3 aforementioned Monte Carlo techniques in their respective sampling performances;

2) To improve the performance of the sampler(s) in sampling from highly-skewed and complex-shaped distributions through the merging of Affine-transformation methods with current algorithm(s). This seeks to address the first research challenge;

3) To develop a sampling algorithm that removes the need by the users to define the tuning parameter of the sampler(s) prior to sampling (i.e. tune-free). This seeks to address the second and third research challenge;

4) To reduce the computational costs and sampling time of the algorithm(s) by reducing the number of hyper-parameters that need to be computed. This seeks to address the third and fourth research challenge;

5) To perform the robust Bayesian identification of key parameters under uncertainty over the likelihood functions. This seeks to address the fifth research challenge;

6) To propose a method to quantify the aleatory and epistemic uncertainties of a set of parameters of a black-box model under model uncertainty. This seeks to address the fifth research challenge;

7) To propose a general approach towards determining the most probable Markov kernel under model uncertainty. This seeks to address the fifth research challenge; and

8) To propose an approach towards merging deterministic model predictions with Bayesian statistics to provide probabilistic predictions of key quantities under sparse data and model uncertainty. This seeks to address the fifth and sixth research challenge.

1.3 Dissertation structure

The dissertation consists of 10 chapters to which an overview is presented in Table 1.1. These chapters have been classified into 3 distinct categories: 1) Literature review; 2) Theoretical development; and 3) Application. In addition, the research objective(s) which is/are to be achieved is also indicated in Table 1.1 along with the corresponding chapter to provide a clear reference. Furthermore, a flow-chart outlining the links between the chapters of the dissertation is presented in Figure 1.1. The dissertation is structured as follows:

Chapter 2 provides a detailed overview of Bayesian model updating and a literature review of its recent applications in engineering. Chapter 3 presents an overview of the aforementioned Monte Carlo approaches along with a literature review of their respective recent applications. In addition, a detailed evaluation and comparison between their sampling performances is provided through numerical and experimental case studies. This seeks to address the first research objective.

Chapter 4 presents the novel Transitional Ensemble Markov Chain Monte Carlo sampling algorithm and investigates in detail the short-comings of the traditional Transitional Markov Chain Monte Carlo sampler. From which, approaches towards addressing these short-comings are discussed and numerical studies are presented to verify the results of the proposed sampler with the traditional Transitional Markov Chain Monte Carlo sampler. This seeks to address the second, third, and fourth research objectives.

Chapter 5 presents the first application of the Transitional Ensemble Markov Chain Monte Carlo sampler in an off-line structural Bayesian identification problem using experimental data. It would incorporate the implementation of the Robust Bayes approach to deal with the uncertainty over the choice of likelihood functions from which a metric is introduced to determine the likelihood with the most and least influence on the posterior results. This seeks to address the fifth research objective.

Chapter 6 presents the second application of the Transitional Ensemble Markov Chain

Monte Carlo sampler to identify uncertain parameters of a dynamic black-box system under uncertainty. This problem is based on the recent NASA-Langley Uncertainty Quantification Challenge 2019 [26] and the focus of the chapter is on the approach towards calibrating a black-box model consisting of both aleatory and epistemic parameters and subsequently reducing its uncertainty. This seeks to address the sixth research objective.

Chapter 7 presents the novel Sequential Ensemble Monte Carlo sampling algorithm and investigates in detail the short-comings of the traditional Sequential Monte Carlo sampler for Sequential Bayesian inference. From which, approaches towards addressing these short-comings are discussed and a numerical study is presented to provide an understanding behind the existence of inherent bounds on the acceptance rates of the sampler and how it can be controlled effectively. This seeks to address the second and third research objectives.

Chapter 8 presents an application of the Sequential Ensemble Monte Carlo sampler in an on-line structural health monitoring problem using experimental data. It seeks to not only validate the proposed sampler in identifying the time-varying model parameter under realistic settings, but also evaluate its robustness in identifying the most probable Markov kernel under model uncertainty. This seeks to address the seventh research objective.

Chapter 9 presents a novel framework aimed at merging Artificial Intelligence tools with Bayesian statistics to produce robust probabilistic estimates of Nuclear material properties under sparse data. To highlight its practicality, experimental data from past campaigns are used to verify and validate the proposed approach. This work serves as a feasibility study submitted towards the recent Game Changers Challenge hosted by the Advanced Nuclear Skills and Innovation Campus in 2022 [54]. This seeks to address the eighth research objective.

Finally, Chapter 10 provides recommendations for the future research works before concluding the dissertation by summarising what has been presented and achieved here.

Chapter	Title	Type	Objective(s)
1	Introduction	–	–
2	Overview of Bayesian Model Updating	Literature review	–
3	Review of the Monte Carlo Techniques	Literature review	1
4	Transitional Ensemble Markov Chain Monte Carlo: A Robust and Efficient sampler for Off-line Bayesian Model Updating	Theoretical development	$2, 3, 4$
5	Robust Off-line Bayesian Identification under Model Uncertainty	Application	5
6	Distribution-free Analysis of a Dynamical Black-box system under Uncertainty	Application	6
7	Sequential Ensemble Monte Carlo: A Robust and Efficient sampler for On-line Bayesian Model Updating	Theoretical development	$2, 3$
8	On-line Identification of Time-varying Model parameters for Structural Health Monitoring	Application	7
9	Probabilistic Prediction of Nuclear Material Properties with Artificial Intelligence under Uncertainty	Application	8
10	Conclusion	–	–

Table 1.1: Content organisation of the dissertation.

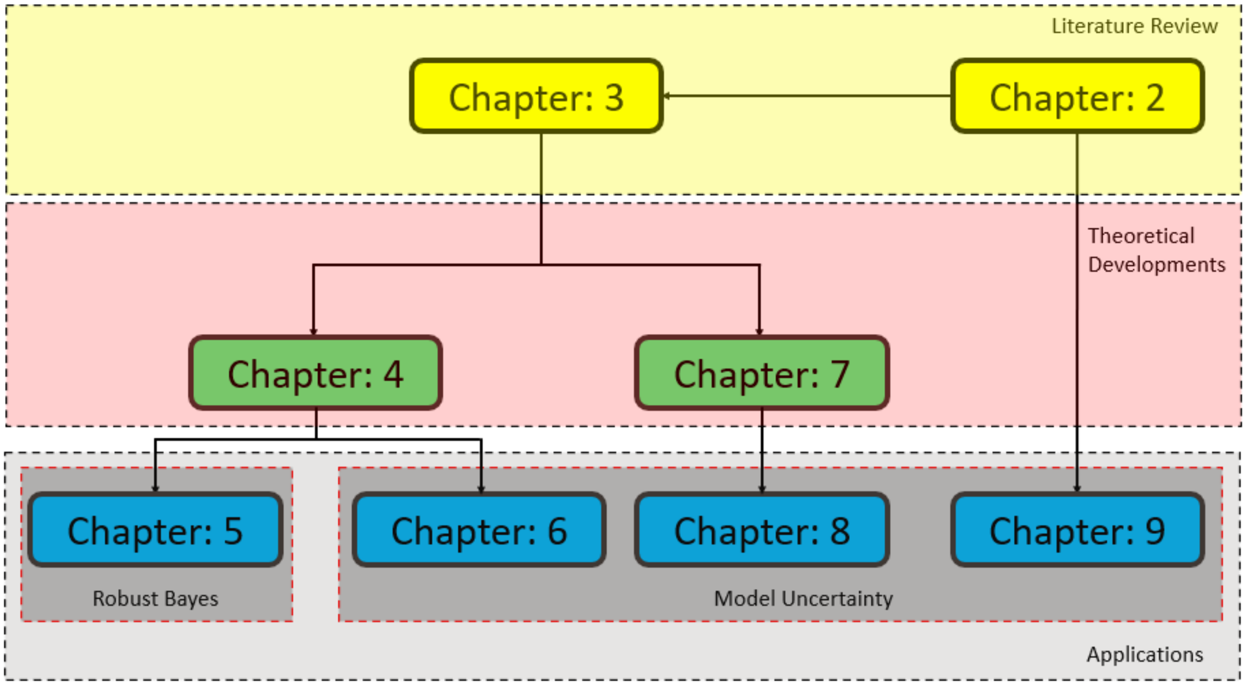


Figure 1.1: Flow-chart illustrating the dissertation structure as well as the links between the chapters.

Chapter 2

Overview of Bayesian Model Updating

Abstract

The chapter provides an overview on the concept of Bayesian inference and model updating which forms the basis of the research works presented in the dissertation. It includes a detailed introduction to the mathematical formalism as well as a literature review of the state-of-the-art developments in this field.

2.1 Probabilistic model updating

Let us consider the problem involving a physical system whose virtual behaviour is modelled by the function $M(\boldsymbol{x}, \boldsymbol{\theta})$ whereby $\boldsymbol{x}$ represents the vector of fixed or unchangeable model parameters, and $\boldsymbol{\theta}$ represents the vector of controllable variables where those values can be changed by the analyst. In general, this function can be linear or non-linear, and it can be used for describing both static or dynamic problems [55].

There exists many mathematical models expressing the relationship between the quantity of interest to be assessed D (e.g. the frequency response function of the system) and the model prediction $M(\boldsymbol{x}, \boldsymbol{\theta})$ [56]. A simple and special case of such is the additive error model [57]:

$$D = M(\boldsymbol{x}, \boldsymbol{\theta}) + \epsilon \tag{2.1}$$

whereby ϵ represents the error caused by measurement errors and/or model parameter uncertainties. The uncertainty in the model parameters $\boldsymbol{\theta}$ can be accounted for using a probability density function. By doing so, one can construct a stochastic model (or a class of models [37]) to obtain a probabilistic prediction on the possible values of the system output D, and therefore its statistics, given the vector of uncertain model parameters $\boldsymbol{\theta}$. These are the so-called forward problems which can be solved by means of analytical approaches [58, 59, 60] or, in general, by Monte Carlo simulation approaches [61, 62, 63]. In the latter approach, the statistics of D are obtained by first generating n realisations of $\boldsymbol{\theta}_i$ (for $i = 1, ..., n$) parameters from a known joint probability distribution $\pi(\boldsymbol{\theta})$. Then, for each realization of $\boldsymbol{\theta}_i$ the model is evaluated to obtain the corresponding realization of D_i (i.e. $D_i = M(\boldsymbol{x}, \boldsymbol{\theta}_i)$). By repeating this process for all n samples, a sample distribution of $\boldsymbol{D}$ is eventually obtained [64, 65].

There are 3 advantages to using the Monte Carlo approach: 1) the Monte Carlo technique is applicable to any problem including non-smooth or non-linear cases (see e.g. [66, 67]); 2) the convergence rate associated with Monte Carlo simulation is independent of the number of random variables making it favourable for solving high-dimensional problem; and 3) the computation performed by the Monte Carlo technique is easily parallelised [67]. Hence, the forward problem implementation is quite simple once the joint distribution of the uncertain variables is defined. However, the real challenge is the identification of the most appropriate joint distribution $\pi(\boldsymbol{\theta})$ that is able to predict some available measurements $\boldsymbol{D}$. This is called the inverse problem [13]. There are 2 main statistical approaches to identify the parameters of a statistical model given a set of observations: 1) the frequentist approach; and 2) the

Bayesian model updating approach. The Maximum likelihood estimator is one of the most often used estimator in the frequentist literature [68, 69, 70, 71] which involves finding the parameter value(s) that maximize(s) the likelihood of observing the $\boldsymbol{D}$ given the parameters $\boldsymbol{\theta}$). The Bayesian model updating approach, on the other hand, casts this inverse problem as a Bayesian inference problem [37, 38, 72] which will be explained in the Section 2.2.

Since the focus of the dissertation is to make inferences on $\boldsymbol{\theta}$, the representation of the model output $M(\boldsymbol{x}, \boldsymbol{\theta})$ can be simplified as $M(\boldsymbol{\theta})$.

2.2 Bayesian inference

A key advantage of adopting Bayesian inference in model updating lies in its ability to combine prior information of a quantity of interest with the observed data to yield a stochastic characterisation of the quantity to be inferred. This yields the posterior distribution of the parameter(s) of interest $\boldsymbol{\theta}$ for a given choice of model $M(\boldsymbol{\theta})$ [37, 38, 73]:

$$P(\boldsymbol{\theta}|\boldsymbol{D}, M) = \frac{P(\boldsymbol{D}|\boldsymbol{\theta}, M) \cdot P(\boldsymbol{\theta}|M)}{P(\boldsymbol{D}|M)} \tag{2.2}$$

whereby

- $\boldsymbol{D}$ represents the vector of the measurements (or observations),
- $P(\boldsymbol{\theta}|M)$ represents the prior distribution,
- $P(\boldsymbol{D}|\boldsymbol{\theta}, M)$ represents the likelihood function of the parameters,
- $P(\boldsymbol{D}|M)$ represents the evidence,
- $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$ represents the posterior distribution, usually the target distribution from which sampling is done.

2.2.1 Prior distribution

The prior distribution, $P(\boldsymbol{\theta}|M)$, is a reflection of one's a priori knowledge or initial hypothesis about the model's parameter(s) to be inferred before any measurements are obtained. It comes in various forms such as expert opinions, lab-scale experiment testing, and previous uncertainty quantification of the parameter(s) of interest [74]. In theory, any type of prior distribution can be used depending on the amount of information available [72].

If what is known about the parameter(s) is/are its upper and lower bounds, then a Uniform distribution could be used as the prior distribution based on the principle of Maximum Entropy [75, 76]. Though this may seem like the most general option, one needs to take note on the selection of the bounds such that the true value(s) of the parameter(s) is/are enclosed within those bounds. An approach would be to choose a significantly large bounds such that the true value(s) is/are included with a high degree of certainty. It is also noteworthy that by adopting the Uniform distribution as the prior, the posterior would simply be proportional to the likelihood function. Some recent research works which adopted the Uniform prior in its Bayesian model updating set-up include: estimating model parameters used to model a bolted structure [77]; structural parameters of a composite structure [78]; crack parameters of a beam structure [79]; stiffness and mass parameters of a DLR-AIRMOD structure [80]; and stiffness parameters of a cantilever beam [81].

On the other hand, if the mean and relative error of the parameter(s) is/are known, then an informative Normal distribution may be used as the prior distribution. Some recent research works which adopted the Normal distribution prior in its Bayesian model updating set-up include: estimating the stiffness parameters within a shear model of a two storey structure [82]; joint-stiffness parameters of the stochastic model for a joint contact surface [83]; the logarithmic ground truth system parameters of a three degrees-of-freedom system [84]; the cosmological parameters used in a supernovae analysis [85]; and state parameters of the dynamical model used for real-time defect detection of high-speed train wheels [86].

For any N_d-dimensional problem, assuming independence between the parameters of interest, $\boldsymbol{\theta}$, the prior distribution can be expressed as follows:

$$P(\boldsymbol{\theta}|M) = P(\boldsymbol{\theta}^1, ..., \boldsymbol{\theta}^{N_d}|M) = \prod_{d=1}^{N_d} P(\boldsymbol{\theta}^d|M) \tag{2.3}$$

whereby $\boldsymbol{\theta}^d$ is the d^{th} dimension (or component) of the vector of input parameters $\boldsymbol{\theta}$.

2.2.2 Likelihood function

The likelihood function, $P(\boldsymbol{D}|\boldsymbol{\theta}, M)$, reflects the degree of error between the obtained measurements, $\boldsymbol{D}$, and the output obtained from the mathematical model $M(\boldsymbol{\theta})$ used to describe the physical system. For the case studies presented in the next chapter, it will be assumed that only one model can be used to describe the observed $\boldsymbol{D}$. In general, there could be multiple models used to represent $\boldsymbol{D}$ and one can associate probabilities to these models to decide the most probable model (i.e. model selection - see [51, 87, 88, 13]). It needs to be pointed out that the model output $M(\boldsymbol{\theta})$ considered in the chapter is purely deterministic. For this reason, the likelihood function has to reflect such error between $\boldsymbol{D}$ and $M(\boldsymbol{\theta})$ for each set of possible $\boldsymbol{\theta}$ values and is derived from an assumed error (i.e. noise) model. As such, the likelihood function is a function of $\boldsymbol{\theta}$ and not of $\boldsymbol{D}$. Assuming that the measurements D_i (for $i = 1, ..., n$) are independently, identically distributed, the likelihood function takes on the following mathematical form:

$$P(\boldsymbol{D}|\boldsymbol{\theta}, M) = \prod_{i=1}^{n} P(D_i|\boldsymbol{\theta}, M) \tag{2.4}$$

Due to the assumption that the error ϵ between the observation and the model follows a zero mean Normal distribution with zero mean and a fixed variance, the common choice of likelihood function would be the Normal distribution [89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100]:

$$P(\boldsymbol{D}|\boldsymbol{\theta}, M) = \left(\prod_{i=1}^{n} \frac{1}{\sigma \cdot \sqrt{2 \cdot \pi}}\right) \cdot \exp\left[-\sum_{i=1}^{n} \frac{(D_i - M(\boldsymbol{\theta}))^2}{2 \cdot \sigma^2}\right] \tag{2.5}$$

As seen in Eq. (2.1), the expression $(D_i - M(\boldsymbol{\theta}))$ simply yields error ϵ_i between the i^{th} measurement and the model output while σ^2 is a hyper-parameter that can be interpreted as the variance of ϵ_i. For cases whereby correlations are present between the measurements D_i, this information would be captured in the non-zero off-diagonal elements in the covariance matrix $\boldsymbol{\Sigma}$ of the Normal likelihood function and the corresponding likelihood function can be written as:

$$P(\boldsymbol{D}|\boldsymbol{\theta}, M) = \frac{1}{\sqrt{|\boldsymbol{\Sigma}| \cdot (2 \cdot \pi)^n}} \cdot \exp\left[-\frac{1}{2} \cdot (\boldsymbol{D} - M(\boldsymbol{\theta}))^{\mathrm{T}} \cdot \boldsymbol{\Sigma}^{-1} \cdot (\boldsymbol{D} - M(\boldsymbol{\theta}))\right] \tag{2.6}$$

There are 3 possible ways to decide on the value of σ: 1) through estimation via the mean squared error of $(D_i - M(\boldsymbol{\theta}))$; 2) by setting σ as an inferred parameter in the Bayesian inference procedure. Such approach is used when there is insufficient knowledge on the measurement error and has been adopted in [101, 3, 102, 103]; and 3) to set σ as a fixed parameter based on prior calculations or knowledge. The strategy commonly adopted is to set σ to correspond to the standard deviation of ϵ, especially if the latter follows a prescribed distribution. For instance, a common choice for ϵ is a zero-mean Normal distribution with standard deviation σ_ϵ, then $\sigma = \sigma_\epsilon$. This strategy will be used in the case studies, presented in the next chapter, to justify the choice of σ for the likelihood function.

It is worth noting that different forms of likelihood functions have been adopted in literature (see e.g. [1, 104, 105, 106, 95, 107, 108]) to capture the degree of agreement between $\boldsymbol{D}$ and $M(\boldsymbol{\theta})$, as summarised in Table 2.1. This choice is related to the assumptions made on the underlying unknown data-generating distribution. For illustration purposes, the likelihood functions listed in Table 2.1 are compared in Figure 2.1 for the case of a mono-dimensional θ. Note that the plot for the lognormal likelihood function is not included in the Figure 2.1 as it is defined in the logarithmic space. Its shape profile, however, follows that of a Normal distribution.

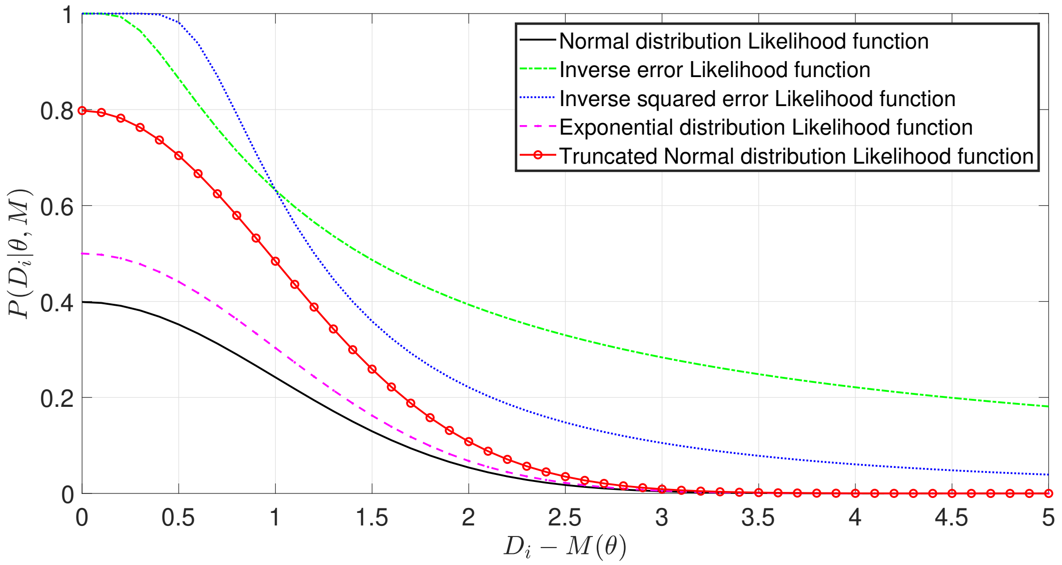


Figure 2.1: Comparison between the different likelihood functions for the case of a mono-dimensional θ. For all the plots, σ is set as 1.

It is important to notice that it is more convenient to use the logarithmic of the likelihood function, called log-likelihood defined as:

$$\log(\boldsymbol{D}|\boldsymbol{\theta}, M)) = \sum_{i=1}^{n} \log(P(D_i|\boldsymbol{\theta}, M)) \tag{2.7}$$

This avoids numerical problems (e.g. arithmetic underflow) with the calculation of the likelihood function. In fact, the calculation of the likelihood requires to compute the product of the likelihood function for each measurement D_i as shown in Eq. (2.4). Another advantage of using the log-likelihood is that the addition operation is much faster to compute than the product operation.

Type	Likelihood function, $P(D_i\|\boldsymbol{\theta}, M)$	Reference(s)
Normal Distribution	$\frac{1}{\sigma \cdot \sqrt{2 \cdot \pi}} \cdot \exp\left[-\frac{(D_i - M(\boldsymbol{\theta}))^2}{2 \cdot \sigma^2}\right]$	[89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100]
Lognormal Distribution	$\frac{1}{\sigma \cdot D_i \cdot \sqrt{2 \cdot \pi}} \cdot \exp\left[-\frac{(\log(D_i) - \log(M(\boldsymbol{\theta})))^2}{2 \cdot \sigma^2}\right]$	[105, 106, 95, 107, 108]
Inverse Error	$1 - \exp\left[-\sqrt{\frac{1}{(D_i - M(\boldsymbol{\theta}))^2}}\right]$	[1]
Inverse Squared Error	$1 - \exp\left[-\frac{1}{(D_i - M(\boldsymbol{\theta}))^2}\right]$	[1]
Exponential Distribution	$\frac{1}{2 \cdot \sigma^2} \cdot \exp\left[-\frac{(D_i - M(\boldsymbol{\theta}))^2}{2 \cdot \sigma^2}\right]$	[104]
Truncated Normal Distribution	$\frac{\sqrt{2}}{\sqrt{\pi} \cdot \sigma} \cdot \exp\left[-\frac{(D_i - M(\boldsymbol{\theta}))^2}{2 \cdot \sigma^2}\right], \text{ for } \|D_i - M(\boldsymbol{\theta})\| \geq 0$	[104]

Table 2.1: Examples of typical likelihood functions used for model updating.

The chapter assumes instances whereby the full likelihood function is known. However, in general, situations can arise whereby the model used is so complex that it becomes computationally expensive to adopt the full likelihood function [109]. In addition, there are instances whereby the model output itself can be stochastic. This will be discussed in Chapter 6 of the dissertation. Under such circumstances, one approach would be to simply adopt the use of Kernel densities to estimate the likelihood function using information from the PDF of the stochastic model output. This method, however, would require a sufficiently large number of model outputs to provide a good estimate of the PDF [109]. Alternatively, one could also turn to the use of approximate likelihood functions and adopt the technique of Approximate Bayesian Computation (ABC) [110, 111]. These approximate likelihood functions capture the discrepancy between the $\boldsymbol{D}$ and the model outputs using stochastic distance metrics such as: Euclidian [112]; Mahalanobis [113]; and Bhattacharyya distances [114]. The implementation of such approximate likelihood functions is also presented in Chapter 6 of the dissertation. For more details on ABC and the various stochastic distance metrics, the readers may refer to the respective references.

2.2.3 Evidence term

The evidence term, $P(\boldsymbol{D}|M)$, serves as the normalising constant of the Bayesian formula to ensure that the posterior (see Eq. (2.2)) integrates to 1. In Bayesian inference, the probability of the observation $P(\boldsymbol{D}|M)$ is fixed and independent of $\boldsymbol{\theta}$. It is therefore a numerical constant. Since the interest is in understanding the relationship between the parameters $\boldsymbol{\theta}$ and the observations $\boldsymbol{D}$, the evidence can be neglected. As such, the resulting proportionality relation for the posterior distribution up to a normalising constant is obtained [102]:

$$P(\boldsymbol{\theta}|\boldsymbol{D}, M) \propto P(\boldsymbol{D}|\boldsymbol{\theta}, M) \cdot P(\boldsymbol{\theta}|M) \tag{2.8}$$

Based on Eq. (2.8), the form of the posterior distribution is only known implicitly.

2.2.4 Posterior distribution

The posterior distribution, $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$, represents the updated distribution of the model parameters, $\boldsymbol{\theta}$, after obtaining some measurements. This reflects the updated knowledge of the model parameters $\boldsymbol{\theta}$ based on the new information obtained from the observations $\boldsymbol{D}$.

There exists analytical solutions for the resulting posterior when the posterior and prior are of the same distribution family. These are known as conjugate distributions [115, 116, 117, 118, 119], and the prior is defined as conjugated for the likelihood function. Typical examples are binomial likelihood and a beta prior for discrete cases and normal likelihood and prior for the continuous cases. However, generally the posterior and prior are non-conjugate distributions. Often, the posterior distribution might not necessarily conform with a well-known parameterized distribution function and, for example, it might be multi-modal. In these situations it would be computationally expensive even if the interest is limited to the analytical evaluation of the mean and variance of the posterior distribution.

The generally applicable numerical technique for estimating distributions is the Monte Carlo method [120]. In particular, Monte Carlo methods can be used to efficiently evaluate the statistics of $\boldsymbol{\theta}$, rather than the full posterior distribution. Suppose one is able to generate samples from $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$, it is possible to estimate the moments of the posterior distribution as follows [121]:

$$\mathrm{E}[\boldsymbol{\theta}] \approx \frac{1}{N} \cdot \sum_{i=1}^{N} \boldsymbol{\theta}_i \tag{2.9}$$

$$\mathrm{E}[\boldsymbol{\theta}^2] \approx \frac{1}{N} \cdot \sum_{i=1}^{N} {\boldsymbol{\theta}_i}^2 \tag{2.10}$$

and the variance estimate of the posterior $\mathrm{Var}[\boldsymbol{\theta}]$ is then computed using:

$$\mathrm{Var}[\boldsymbol{\theta}] = \mathrm{E}[\boldsymbol{\theta}^2] - \left(\mathrm{E}[\boldsymbol{\theta}]\right)^2 \tag{2.11}$$

whereby $\mathrm{E}[\boldsymbol{\theta}]$ and $\mathrm{E}[\boldsymbol{\theta}^2]$ are obtained from Eq. (2.9) and Eq. (2.10) respectively. Marginals and quantiles of the distribution can also be computed using the same realisations.

As seen in Eq. (2.8), the posterior is only known implicitly up to a normalisation constant. To sample from the posterior in this case, the Markov Chain Monte Carlo (MCMC) [50] and the Sequential Monte Carlo (SMC) [52] samplers are the 2 most implemented approaches, and that the Transitional Markov Chain Monte Carlo (TMCMC) [51] sampler is a particular variant of the SMC sampler. Details on the respective sampling techniques will be discussed in Chapter 3.

The above-mentioned techniques can be used to construct a Markov chain on the model parameters space $\boldsymbol{\theta}$ whose steady state distribution is the posterior distribution of interest $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$ [122]. MCMC does not require the evaluation of the evidence, and Eq. (2.8) can be used directly. MCMC only requires evaluation of the joint distribution of Eq. (2.8) up to a proportionality factor and point-wise for any generated sampled of $\boldsymbol{\theta}$. Therefore, these sampling algorithms return samples $\boldsymbol{\theta_i}$ (for $i = 1, \ldots, N$) where each sample can be assumed to be drawn from $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$.

2.3 Applications of Bayesian model updating

The technique of Bayesian model updating has been adopted in many applications, for instance: to quantify the discrete element methods prediction of the behavior of granular materials [123]; to update the probabilistic model related to the boundary condition and to estimate torsional stiffness parameter of a cantilever beam under uncertainty through vibrational analysis [81]; in structural health monitoring by identifying the position and severity of a crack in a suspension arm of a car [1]; to update the material dependent constants of the Paris-Erdogan Law used to predict crack growth rate in a carbon-steel Nuclear piping [124]; to perform on-line estimation of parameters of building energy models based on information from in-situ sensor [125]; and to estimate the most probable leakage scenarios for the purpose of leakage detection in water distribution networks [126].

Details on the Bayesian model updating set-up in these references are summarised in Table 2.2. More recently, the technique of Bayesian model updating has also been developed to include elements of structural reliability, giving rise to Bayesian Updating with Structural Reliability (BUS) methods [127]. This, however, will not be discussed given that it involves the use of structural reliability methods which is beyond the scope of the dissertation.

Application	Prior	Likelihood(s)	Sampling technique
Estimate torsional stiffness parameter for a cantilever beam [81]	Uniform	Normal	MCMC
Identify the material dependent constants of the Paris-Erdogan Law for crack growth rate prediction [124]	Normal	Normal	MCMC
Quantify the discrete element methods prediction of granular materials' behaviour [123]	Uniform	Normal	TMCMC
Leakage detection in water distribution networks [126]	Uniform	Normal	TMCMC
On-line monitoring (crack detection) in a suspension arm of a car [1]	Uniform	Normal, Inverse Error, Inverse Squared Error	TMCMC
On-line parameter estimation of building energy models [125]	Normal	Normal	SMC

Table 2.2: Summary of Bayesian Model Updating set-up and sampling technique employed.

Chapter 3

Review of the Monte Carlo Techniques

Abstract

The chapter provides a review of the advanced Monte Carlo sampling methods in the context of Bayesian model updating within engineering applications: 1) Markov Chain Monte Carlo; 2) Transitional Markov Chain Monte Carlo; and 3) Sequential Monte Carlo samplers. This includes details on the respective algorithms, a literature review of their recent implementations within the engineering field, and an evaluation on the advantages and disadvantages of the respective sampling techniques. To provide a comparison in the sampling performance between each of these sampling techniques, the chapter presents 3 case studies of increasing complexity. The first case study presents the parameter identification for a spring-mass system under a static load. The second case study presents a 2-dimensional bi-modal posterior distribution and the aim is to observe the performance of each of these sampling techniques in sampling from such distribution. Finally, the last case study presents the stochastic identification of the model parameters of a complex and non-linear numerical model based on experimental data. For each of the case studies, the numerical implementations and parametric settings of the respective samplers are provided.

3.1 Markov Chain Monte Carlo

3.1.1 Conceptual introduction

The MCMC sampler is a sampling technique introduced by Metropolis [128] which encompasses 2 main aspects: Monte Carlo simulations and Markov chains. The concept of Markov chains was devised by Andrey Markov in 1906 and it refers to a sequence of random samples (or states) θ_i, for $i = 1, 2, ..., N$, whereby the value of θ_{i+1} depends only on the previous value θ_i [129]. This is also known as the Markov property [130]. A Markov chain initiates from θ_1 and from there, the transition between successive samples in the chain (i.e. from θ_i to θ_{i+1}) would occur with probability $T(\theta_i \rightarrow \theta_{i+1})$ known as the transition probability [131] which is determined by a transition probability distribution function. Through this Markov process, it is assumed that the distribution of the generated samples θ_i would converge to a stationary distribution. However, the initial samples of the Markov chain are generally not distributed according to the stationary distribution and thus, not representative of the stationary distribution. To address this, one can discard the initial $N_{burn-in}$ number of samples. This procedure is known as the burn-in and $N_{burn-in}$ corresponds to the burn-in length of the Markov chain [132]. In the context of Bayesian model updating, this stationary distribution corresponds to $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$.

There are many variants of MCMC techniques currently in existence and 2 of the most commonly used variants are: 1) the Gibbs sampler [133, 134]; and 2) the Metropolis-Hastings (MH) sampler [50]. A brief description to the Gibbs sampler is provided as follows: The Gibbs sampler is a special instance of the MH sampler and is implemented to sample from a N_d-dimensional multi-variate posterior $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$. Instead of sampling directly from $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$, the sampling algorithm does so indirectly through the use of the conditional probability distribution of a d^{th} component of $\boldsymbol{\theta}$ (i.e. $\boldsymbol{\theta}^d$) conditioned on the remaining $N_d - 1$ components of $\boldsymbol{\theta}$. The basic assumptions behind the Gibbs sampler are that such conditional distributions for all $\boldsymbol{\theta}^d$ are known and that they are relatively easier to sample from [135]. Let the conditional probability distribution for $\boldsymbol{\theta}^d$ be denoted as $P(\boldsymbol{\theta}^d|\{\boldsymbol{\theta}^1, \ldots, \boldsymbol{\theta}^{d-1}, \boldsymbol{\theta}^{d+1}, \ldots, \boldsymbol{\theta}^{N_d}\}, \boldsymbol{D}, M)$. Through such approach, the sampling algorithm is able to generate samples from $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$ in a component-wise manner whereby at a given sampling iteration, each $\boldsymbol{\theta}^d$ is being sampled one at a time in a sequential manner through its respective conditional distribution [135]. To provide an understanding of the sampling procedure, the algorithmic description of the Gibbs sampler in sampling N samples from a general N_d-dimensional posterior is presented in Algorithm 1. It needs to be noted that

Algorithm 1 describes the systematic scan Gibbs sampler. In practice, the random scan Gibbs sampler is more commonly implemented.

Algorithm 1 Gibbs sampling algorithm (N_d-dimensional case)

```
procedure (Generate samples from a general $N_d$-dimensional posterior.)
    Draw initial sample set: $\boldsymbol{\theta}_1 = (\theta_1^1, ..., \theta_1^d) \sim P(\boldsymbol{\theta}|M)$    ▷ Initialise chain
    for $i = 1 : N - 1$ do    ▷ Generate Markov chain samples
        for $d = 1 : N_d$ do    ▷ Sample the $d^{th}$ component
            Draw sample: $\theta_{i+1}^d \sim P(\boldsymbol{\theta}^d|\{\theta_{i+1}^1, \ldots, \theta_{i+1}^{d-1}, \theta_i^{d+1}, \ldots, \theta_i^{N_d}\}, \boldsymbol{D}, M)$
        end for
    end for
end procedure
```

A problem with the Gibbs sampler, however, is in the selection of an appropriate conditional probability distribution for each $\boldsymbol{\theta}^d$ [136]. In general, this may not be trivial because $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$ may be functionally complex. Under such condition, it becomes difficult to derive the conditional distribution for each component $\boldsymbol{\theta}^d$. This limits the implementation of the sampling algorithm, making the Gibbs sampler less general. Hence, in this regard, the MH sampler variant of MCMC will be discussed in detail and implemented to address the problems presented in the chapter. One key strength of the MH algorithm which motivated its implementation, is in its ability to sample from any probability distribution as long as the function that is proportional to its actual normalised density (i.e. $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$ in the form of Eq. (2.8)) is known and that the values of that function can be computed [50].

Without the loss of generality, this section will first elaborate the steps of the MH algorithm for sampling from a one-dimensional posterior before generalising to the case of sampling from a multi-dimensional posterior. The MH sampler is a random-walk algorithm that provides a selection criteria to which the samples are chosen during the sampling procedure. This is done through the use of a proposal distribution $q(\theta^*|\theta_i)$ to generate the next candidate sample θ^* of the chain from a known and relatively simpler distribution from the current sample θ_i. It should be noted that the choice of $q(\theta^*|\theta_i)$ is such that its density function is strictly positive across the entire sample space for which the $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$ is defined. A typical choice of $q(\theta^*|\theta_i)$ is the Normal distribution [137], although it has also been argued in [138] that the selection of an optimal $q(\theta^*|\theta_i)$ is often made on an ad-hoc basis. From there, the generated samples are accepted or rejected based on a given acceptance rule. Figure 3.1 illustrates graphically the principle of the MH sampler: From the current sample θ_i, a candidate sample of the Markov chain, θ^*, is sampled from the proposal distribution

$q(\theta^*|\theta_i)$. Next, the candidate sample θ^* is accepted with probability α defined as:

$$\alpha = \min\left[1, \frac{P(\theta^*|\boldsymbol{D}, M)}{P(\theta_i|\boldsymbol{D}, M)} \cdot \frac{q(\theta_i|\theta^*)}{q(\theta^*|\theta_i)}\right] \tag{3.1}$$

whereby $P(\theta^*|\boldsymbol{D}, M)$ represents the posterior value evaluated at the candidate sample θ^*, and $P(\theta_i|\boldsymbol{D}, M)$ represents the posterior value evaluated at θ_i. $q(\theta_i|\theta^*)$ represents the probability of sampling θ_i given that the current sample is θ^*, and $q(\theta^*|\theta_i)$ represents the probability of sampling θ^* given that the current sample is θ_i as determined by $q(\theta^*|\theta_i)$. Substituting in the posterior distribution in Eq. (3.1) with its definition from Eq. (2.2), one obtains:

$$\alpha = \min\left[1, \frac{P(\boldsymbol{D}|\theta^*, M) \cdot P(\theta^*|M)/P(\boldsymbol{D}|M)}{P(\boldsymbol{D}|\theta_i, M) \cdot P(\theta_i|M)/P(\boldsymbol{D}|M)} \cdot \frac{q(\theta_i|\theta^*)}{q(\theta^*|\theta_i)}\right] \tag{3.2}$$

From Eq. (3.2), it can be seen that the normalisation constant $P(\boldsymbol{D}|M)$ is cancelled out. This further justifies why there is no need to evaluate $P(\boldsymbol{D}|M)$ and that the computation of Eq. (3.1) can be done using an un-normalised posterior (see Eq. (2.2)). This allows the MH algorithm to perform sampling on such distributions. For the case whereby $q(\theta^*|\theta_i)$ is a symmetrical function (i.e. Normal distribution) that is centered about θ_i such that $q(\theta^*|\theta_i) = q(\theta_i|\theta^*)$, the acceptance probability α in Eq. (3.1) becomes:

$$\alpha = \min\left[1, \frac{P(\theta^*|\boldsymbol{D}, M)}{P(\theta_i|\boldsymbol{D}, M)}\right]. \tag{3.3}$$

What Eq. (3.3) implies is that the candidate sample θ^* is always accepted if the samples are moving towards the region of high probability density (i.e. $\frac{P(\theta^*|\boldsymbol{D},M)}{P(\theta_i|\boldsymbol{D},M)} > 1$), otherwise it is accepted with probability α. In practice, a random number r is sampled from a Uniform distribution ranging between 0 and 1 (i.e. $r \sim U[0, 1)$). If $\alpha \geq r$, the proposed sample θ^* is accepted (i.e. $\theta_{i+1} = \theta^*$). Otherwise, θ^* is rejected (i.e. $\theta_{i+1} = \theta_i$).

The characteristics of Markov chains are, thus, ideal to generate samples from the unknown posterior distribution. As a result of the use of Markov chains, it makes the MCMC algorithm inherently serial in its computations. This however does not imply that the computations cannot be parallelised. In fact, there have been developments made in recent years to achieve this as seen in the works by Wilkinson (2005) [139] and Brockwell (2006) [140], though efforts have proved to be non-trivial. Besides the MH algorithm, there are algorithms to other variants of the MCMC methods. These include: 1) Slice sampling [141]; 2) Hamiltonian Monte Carlo methods [49, 142]; 3) Metropolis-adjusted Langevin algorithm [48, 143, 144, 142]; 4) Multiple-try Metropolis [145, 146]; 5) Reversible-jump MCMC [147];

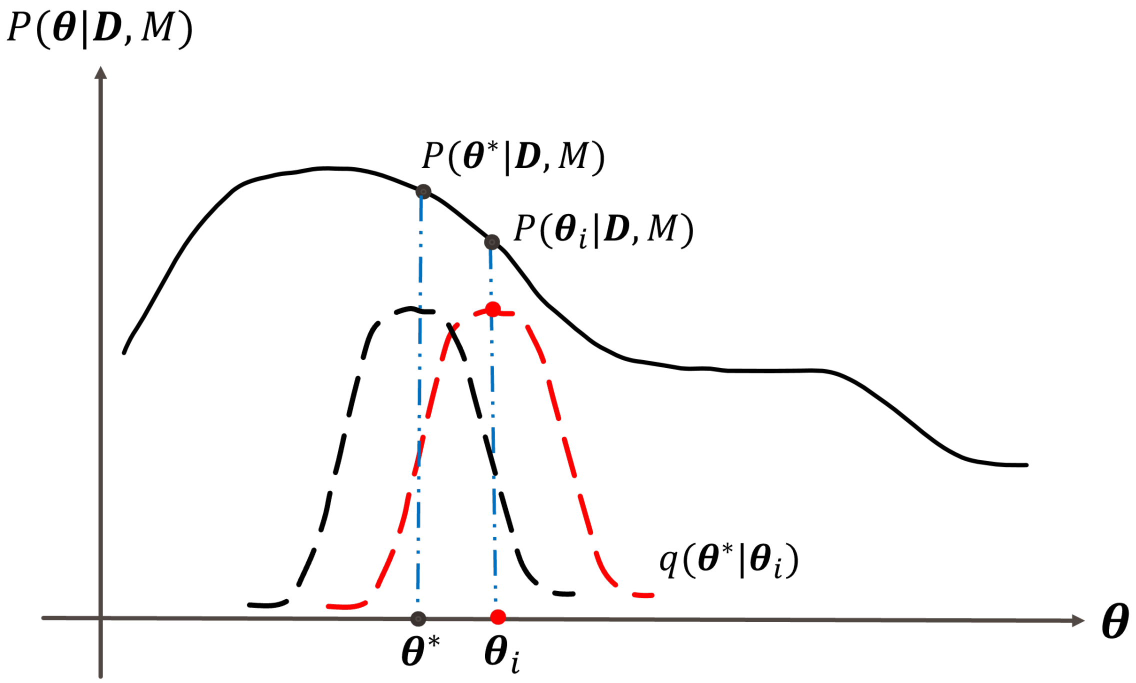

$P(\boldsymbol{\theta}|\boldsymbol{D}, M)$
$P(\boldsymbol{\theta}^*|\boldsymbol{D}, M)$
$P(\boldsymbol{\theta}_i|\boldsymbol{D}, M)$
$q(\boldsymbol{\theta}^*|\boldsymbol{\theta}_i)$
$\boldsymbol{\theta}$
$\boldsymbol{\theta}^*$
$\boldsymbol{\theta}_i$

Figure 3.1: Metropolis-Hasting sampling - the proposal distribution $q(\theta^*|\theta_i)$ (red dotted curve); posterior $P(\theta|\boldsymbol{D}, M)$ (black solid curve); current sample θ_i; and proposed sample θ^*.

and 6) Pseudo-marginal Metropolis-Hastings algorithm [148]. Details on each of these algorithms can be found in their respective references.

In the case studies, the Normal distribution will be used as the proposal distribution with mean defined by the value of the current sample θ_i and standard deviation σ_p that serves as the tuning parameter of the MH algorithm. The choice of the tuning parameter is an important consideration when implementing the algorithm as this will have an impact on the efficiency of the MCMC sampling process. To illustrate the effects of the tuning parameter on the sampling process, the MH sampler is implemented to generate $N = 1000$ samples from a posterior defined by the Normal distribution with mean value of 170 and a standard deviation value of 5. For this example, 3 different values of σ_p for the proposal distribution is used: 100.5; 23.5; and 1.5. The resulting trace plots of the generated samples given the respective values of σ_p are presented in Figure 3.2. To measure the degree of serial correlation between the samples which are drawn by the MH sampler, the autocorrelation function plot is generated for each value of σ_p and presented in Figure 3.2 [149]. In generating the autocorrelation function plots, the sample set is divided across 40 equal time lags from which the autocorrelation coefficient between each time lag is computed.

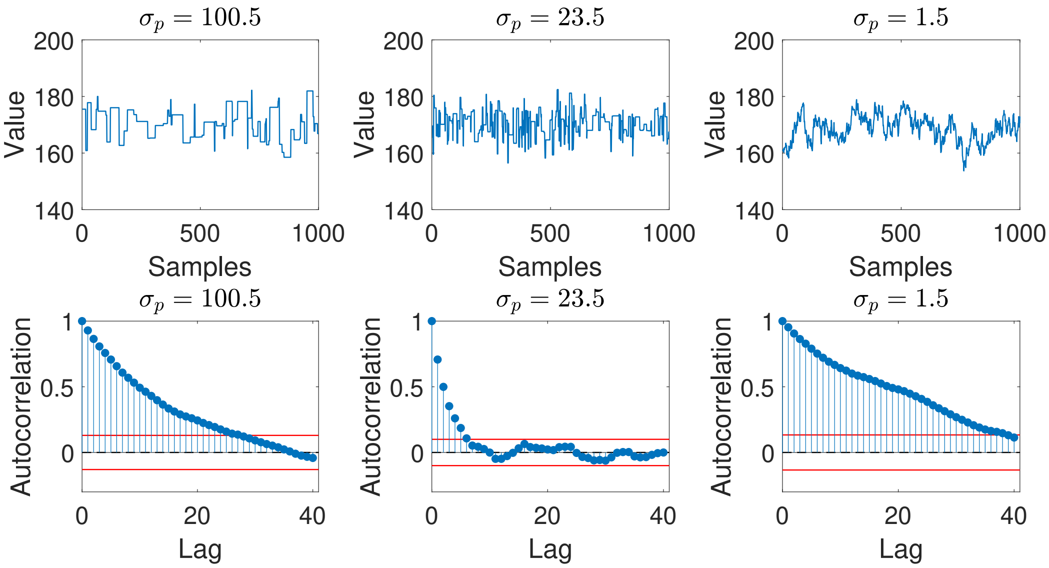


Figure 3.2: The corresponding trace plots and autocorrelation plots obtained with $N = 1000$ samples using: 1) $\sigma_p = 100.5$ yielding an acceptance rate of 0.069; 2) $\sigma_p = 23.5$ yielding an acceptance rate of 0.234; and 3) $\sigma_p = 1.5$ yielding an acceptance rate of 0.896.

From Figure 3.2, the following observations are made: In the case of $\sigma_p = 1.5$, it leads to a relatively smaller jump-size between a sample and the next successive sample of the chain. This gives rise to a relatively high acceptance rate of 0.896 given that next successive sample of the chain is close to the current sample of the chain that the computed value of α (i.e. see Eq. (3.1)) is close to 1. As a result, there is a low probability there are repetitions between successive samples of the chain as indicated by the trace plot. However, such choice of σ_p also gives rise to a high serial correlation between successive samples of the chain as indicated in the corresponding autocorrelation function plot. From the autocorrelation function plot, it can be seen that not only is the autocorrelation between the samples of the chain relatively higher at any given time lag, but the autocorrelation coefficient decreases and converges close to 0 at a slow rate, implying that more sample draws are required ensure the mixing of the samples of the Markov chain.

On the other hand, in the case of $\sigma_p = 100.5$, it leads to a relatively larger jump-size between a sample and the next successive sample of the chain. When this happens, it gives rise to a relatively low acceptance rate of 0.069 (i.e. close to 0) given that in such case, many of the proposed samples may end up lying outside the range of the posterior which leads the computed value of α being close to 0. This leads to a high rejection rate of the proposed samples and a higher probability of repeated samples along the Markov chain as indicated by the trace plot. While the corresponding autocorrelation function plot indicates a relatively quicker rate of convergence of the autocorrelation coefficient of the samples to 0, this is still not favourable as such low acceptance rate reflects the poor exploration of the sample space defined by the posterior which requires more samples to be drawn to ensure that the distribution of the samples converges to the posterior.

Under specific condition, the optimal value of σ_p is the one which produces an acceptance rate around 0.234 [150]. Such acceptance rate is achieved in the case of $\sigma_p = 23.5$ in this illustrative example. From the trace plot, it can be observed that the samples of the Markov chain have converged towards a stationary distribution. In addition, the corresponding autocorrelation function plot indicated that the the number of time lags taken for the autocorrelation coefficient to reach 0 is the shortest. This is favourable given that it indicates that less sample draws are required for the serial correlation between the samples of the Markov chain to converge to (or close to) 0 and therefore, exhibiting a relatively higher mixing rate [149].

To further justify the optimality of having an acceptance rate close to 0.234, the Monte Carlo standard error σ_{SE} is used as the statistical performance metric and is computed from the MH samples given each choice of the tuning parameter σ_p following [151]:

$$\sigma_{SE} = \frac{\sigma_\theta}{\sqrt{N}} \tag{3.4}$$

Here, σ_θ is the Monte Carlo standard deviation which is estimated using the overlapping batch means estimator [151, 152]:

$$\sigma_\theta \approx \frac{N b_n}{(N - b_n)(N - b_n + 1)} \cdot \sum_{y=1}^{N-b_n+1} \left(\bar{\theta}_y - E[\theta]\right) \tag{3.5}$$

where N is the sample size obtained from the posterior, n_b is the batch size, $\bar{\theta}_y = \frac{1}{n_b} \cdot \sum_{g=0}^{b_n-1} \theta_{y+g}$ is the batch mean of the given y^{th} sample batch, and $E[\theta]$ is the sample mean computed using Eq. (2.9). In this illustrative example, the batch size is set at $n_b = 100$ and the resulting values of σ_{SE} for each given choice of σ_p are presented in Table 3.1. From the table, it can be seen that given a finite sample size of $N = 1000$, the Monte Carlo standard error of the estimates is the lowest in the case of $\sigma_p = 23.5$ which corresponds an acceptance rate of 0.234. Such result provides a motivation to use the acceptance rate value of 0.234 as a reference for the implementation of the MH sampler in the subsequent case studies presented later in the chapter.

σ_p	100.5	23.5	1.5
Acceptance rate	0.069	0.234	0.896
σ_{SE}	0.643	0.273	1.727

Table 3.1: Numerical results of the Monte Carlo standard error σ_{SE} obtained for each given choice of tuning parameter σ_p.

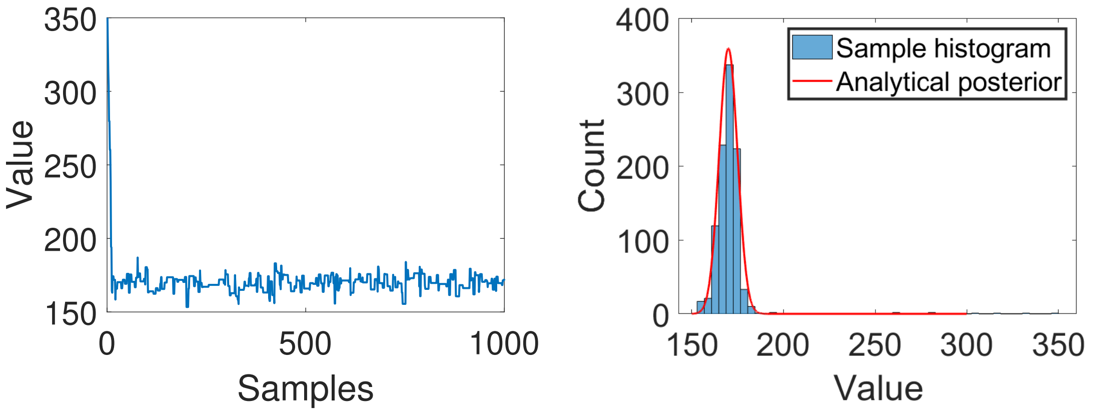
It needs to be noted, however, that the acceptance ratio is a trade-off between making too many small accepted steps and making too many large proposals that get rejected [153]. In fact, for any value of acceptance rate between 0.15 to 0.50, the efficiency of the algorithm is still at least 80 % [149]. Alternatively, to avoid the need to consider the acceptance rate, one could turn to rejection-free and tune-free MCMC algorithms to which examples include: 1) Gibbs sampling [154]; 2) Slice sampling [141]; and 3) the Adaptive Metropolis algorithm [155].

Another important aspect in the sampling procedure by the MH sampler is the need to consider the burn-in length of the chain. This can be checked and inspected visually through constructing a trace plot and identifying the sample number at which the plot begins to converge [132]. As an illustration, the same set-up from the earlier example is used to generate Figure 3.2. From the posterior, $N = 1000$ samples will be generated via MH sampling with a burn-in length of 0. The starting value of the chain is randomly sampled from a Uniform distribution with bounds between 0 and 400. This practice of randomly selecting θ_1 from the prior will be adopted in all the problems presented in this tutorial. The resulting trace plot and histogram are provided in Figure 3.3. From the trace plot in Figure 3.3(a), it can be observed that the plot starts to converge after 15 samples are obtained indicating that $N_{burn-in} = 15$. Figure 3.4 illustrates the resulting trace plot and histogram profile after accounting for burn-in.

3.1.2 Algorithmic description

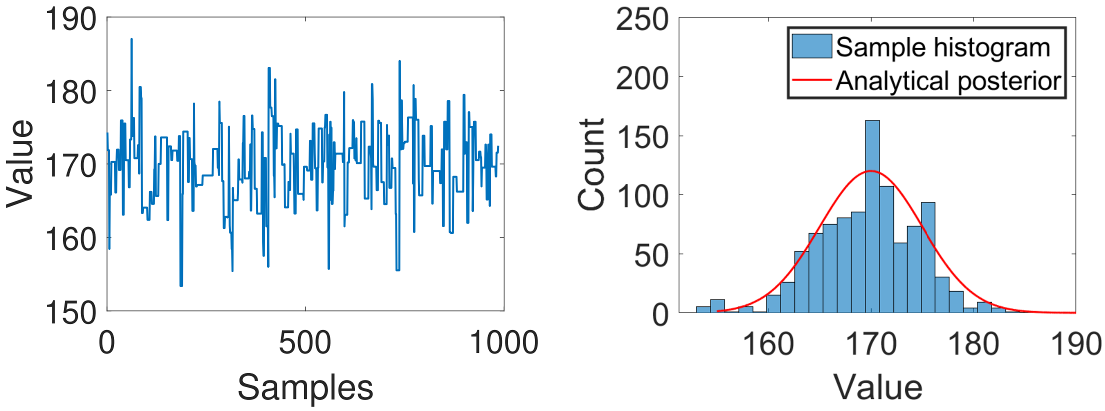
The MH algorithm to generate samples from a one-dimensional posterior is summarised as follows:

1. Set $i = 1$; sample $\theta_i \sim P(\theta)$;



(a) (b)

Figure 3.3: Example of the resulting trace plot (a) and histogram (b) obtained from MH sampling of $N = 1000$ samples with $N_{burn-in} = 0$.



(a) (b)

Figure 3.4: The resulting trace plot (a) and histogram (b) obtained from MH sampling of $N = 1000$ samples with $N_{burn-in} = 15$.

2. Generate candidate sample $\theta^* \sim q(\theta^*|\theta_i)$;

3. Evaluate the posterior distribution at the proposed sample (i.e. $P(\theta^*|\boldsymbol{D}, M)$);

4. Compute the acceptance ratio, α, from Eq. (3.3).

5. Sample $r \sim U(0, 1]$. If $\alpha \geq r$, set $\theta_{i+1}{=}\theta^*$. Otherwise, set $\theta_{i+1}{=}\theta_i$;

6. Set $i = i + 1$ and repeat steps (2) to (6) until termination criteria is met (i.e. total sample size of chain obtained, or stability of the distribution is achieved).

The MH sampler can be generalised to sample from a multi-dimensional posterior. There are 2 ways in which this can be done: 1) block-wise; and 2) component-wise. In the block-wise approach, the proposal distribution $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}_i)$ is a multi-variate function with the same dimensionality as the posterior. Candidate samples are generated from across multiple dimensions at the same time [156, 157]. In essence, variables across all dimensions are updated simultaneously rather than sequentially as per the case in component-wise approach. An example of a sampling method which utilises the block-wise sampling approach would be the Metropolis within Gibbs algorithm [158]. A key problem with this approach is that the acceptance rate drops with the increasing dimensionality of the problem. This is because as the dimension of the posterior increases, it becomes more difficult to determine a suitable $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}_i)$ due to the increased complexity of the entire sample space, especially if the posterior is highly-anisotropic across dimensions. In the component-wise approach, sampling is performed independently for each dimension and variables are updated one dimension at a time in a serial manner [159, 160, 161]. The proposal distribution can be uni-variate or multi-variate, the latter taking the form $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}_i) = \prod_{d=1}^{N_d} q(\theta^{d*}|\theta_i^d)$ whereby d denotes the dimension (or component) number while N_d denotes the total dimension of the $\boldsymbol{\theta}$. It should be pointed out that the workings of the Gibbs sampler is analogous to the component-wise MH sampling [154]. In the chapter, the component-wise approach is adopted to sample from a multi-variate posterior while a multi-variate Normal distribution would be used as the choice for $q(\boldsymbol{\theta^*}|\boldsymbol{\theta_i})$ with covariance matrix $\boldsymbol{\Sigma}_p$. For such case, the covariance matrix $\boldsymbol{\Sigma}_p$ serves as the tuning parameter of the sampler and takes the form of a diagonal square matrix whose non-diagonal matrix elements are 0.

To provide a simple illustration, an explanation is first provided for the case of sampling from a 2-dimensional posterior. For each iteration i, the updating procedure is such that the first component θ^1 is updated first whilst keeping the second component θ^2 constant before the same procedure is repeated for θ^2 whilst keeping the already updated component θ_1 constant. In addition, for a given i^{th} sample, a convention is used whereby $\boldsymbol{\theta}_i^1 = (\theta_i^1, \theta_i^2)$ denotes the first state vector in the current iteration whilst $\boldsymbol{\theta}^{1*} = (\theta^{1*}, \theta_i^2)$ is the first proposed state vector. Similarly, $\boldsymbol{\theta}_i^2 = (\theta_{i+1}^1, \theta_i^2)$ is the second state vector in the current iteration whilst $\boldsymbol{\theta}^{2*} = (\theta_{i+1}^1, \theta^{2*})$ is the second proposed state vector. Thus, extending this convention to a general N_d-dimensional case, the notation $\boldsymbol{\theta}_i^d$ (for $d = 1, ..., N_d$) is used to represent the d^{th} state vector of the samples in the current iteration i and $\boldsymbol{\theta}^{d*}$ to represent the updated d^{th} proposed state vector of the samples in the current iteration i. Here, $\boldsymbol{\theta}_i^d = (\theta_{i+1}^1, ..., \theta_{i+1}^{d-1}, \theta_i^d, \theta_i^{d+1}, ..., \theta_i^{N_d})$ while $\boldsymbol{\theta}^{d*} = (\theta_{i+1}^1, ..., \theta_{i+1}^{d-1}, \theta^{d*}, \theta_i^{d+1}, ..., \theta_i^{N_d})$. Using this generalised convention, the algorithmic description of the MH sampler in sampling from a general N_d-dimensional posterior is presented in Algorithm 2.

Algorithm 2 Component-wise MH algorithm (N_d-dimensional case)

```
procedure (Generate samples from a general d-dimensional posterior.)
    Draw initial sample set: $\boldsymbol{\theta}_1 = (\theta_1^1, ..., \theta_1^d) \sim P(\boldsymbol{\theta}|M)$    ▷ Initialise chain
    for $i = 1 : N - 1$ do    ▷ Generate Markov chain samples
        for $d = 1 : N_d$ do    ▷ Update $d^{th}$ component
            Draw candidate sample: $\theta^{d*} \sim q(\theta^{d*}|\theta_i^d)$
            $\alpha_d = \min\left[1, \frac{P(\boldsymbol{\theta}^{d*}|\boldsymbol{D},M)}{P(\boldsymbol{\theta}_i^d|\boldsymbol{D},M)}\right]$
            Sample: $r_d \sim U(0,1]$
            if $\alpha_d \geq r_d$ then
                $\theta_{i+1}^d = \theta^{d*}$
            else
                $\theta_{i+1}^d = \theta_i^d$
            end if
        end for
    end for
end procedure
```

3.1.3 Applications

Currently, the MH sampler has already been implemented in numerous applications to which examples include: 1) predicting precipitation behaviours in Nickel-Titanium alloys [162]; 2) analysing an electrochemical impedance spectra and estimating the conductivity of a Lithium ion within a solid-state oxide electrolyte [163]; 3) identification of structural damage based on time domain data [164]; 4) to sample classical thermal states from one-dimensional Bose-Einstein quasi-condensates under the classical fields approximation [165]; 5) updating the finite element model of a concrete structure [166]; 6) quantifying the uncertainty associated with the joint model parameters of a stochastic generic joint model [83]; 7) performing joint input-state-parameter estimation for wave loading [84]; 8) performing Bayesian system identification of dynamical systems [167]; and 9) perform Bayesian model identification of higher-order frequency response functions of structures [77].

3.2 Transitional Markov Chain Monte Carlo

3.2.1 Conceptual introduction

The TMCMC sampler is based on the adaptive Metropolis-Hastings MCMC technique [168] whereby samples are not obtained directly from a complex posterior distribution, but rather from a series of relatively simpler "transitional" distributions. The key difference in the sampling procedure between the MCMC and the TMCMC technique is that the MCMC samples are obtained through one or few, successive (very) long Markov chains of length N, whereas TMCMC samples are obtained through N independent Markov chains. This method of obtaining samples is useful especially in cases when the shape of the posterior distribution is complex such as having multiple sharp peaks. The transitional distributions are defined as such [51]:

$$P^{j} \propto P(\boldsymbol{D}|\boldsymbol{\theta}, M)^{\beta_j} \cdot P(\boldsymbol{\theta}|M) \tag{3.6}$$

Here, j denotes the transition step number taking values between 0 to j_{end}, where j_{end} denotes the last iteration number. β_j is the tempering parameter which takes values such that $\beta_0 = 0 < \beta_1 < ... < \beta_{j_{end}-1} < \beta_{j_{end}} = 1$. This allows for the transitional distribution to transit from the prior to the posterior distribution (i.e. $P^0 = P(\boldsymbol{\theta}|M)$ to $P^{j_{end}} = P(\boldsymbol{\theta}|\boldsymbol{D}, M)$). As an illustrative example, the TMCMC sampler is implemented to sample from a one-dimensional posterior defined by a Uniform distribution prior with bounds -4 to 4, and a likelihood function in the form of a Normal distribution with mean 0 and standard deviation 0.105. Figure 3.5 depicts the resulting series of analytical plots which show the evolution of a one-dimensional transitional distribution from the Uniform prior the final posterior.

j	0	1	2	3	4
β_j	0	0.0084	0.0700	0.5459	1

Table 3.2: Numerical values of β_j of the transitional distributions for the corresponding iteration j shown in Figure 3.5.

To provide an understanding of the workings behind the TMCMC sampler, an explanation to its procedure in sampling from a one-dimensional posterior will first be provided. In practice, at transition step $j = 0$ (i.e. $\beta_j = 0$), N samples are generated from the prior via direct random sampling using the Monte Carlo method. For transition steps $j \geq 1$ (i.e. while $\beta_j < 1$), the tempering parameter β_j is computed. From there, the transitional distribution

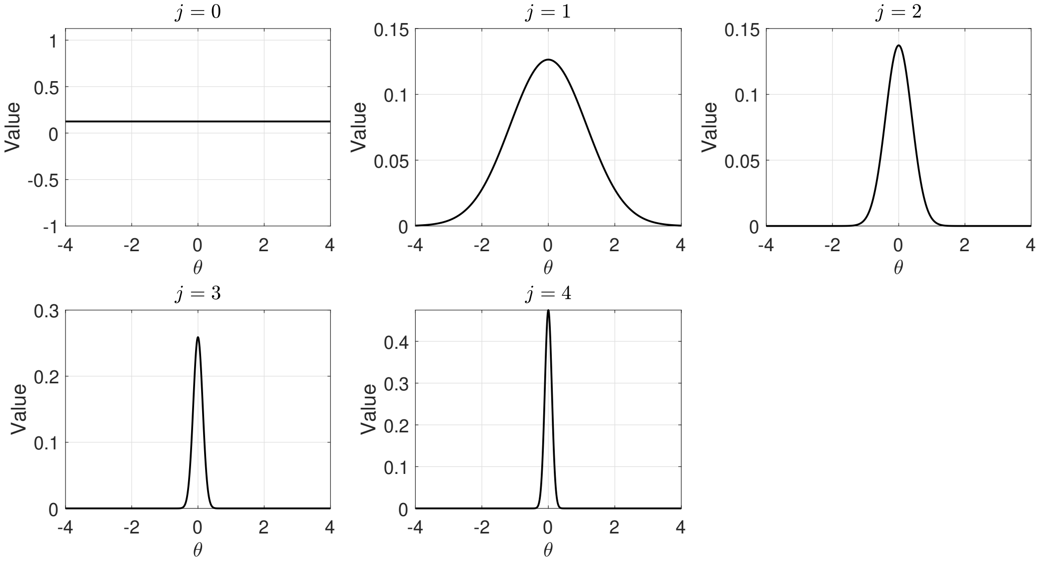


Figure 3.5: Evolution of the transition distribution from an initial Uniform prior distribution to the final posterior distribution across the iterations j.

P^j is defined using Eq. (3.6) and N samples are then obtained from P^j using MH sampler through the following procedure: First, a statistical weight function $\hat{w}_i^j$ is determined to describe the statistical (or importance) weight associated with each sample θ_i (for $i = 1, \ldots, N$) in a given iteration j. This statistical weight function $\hat{w}_i^j$ is mathematically defined in Eq. (3.7) as:

$$\hat{w}_i^j = \frac{P(\boldsymbol{D}|\theta_i, M)^{\Delta\beta_j}}{\sum_{i=1}^{N} P(\boldsymbol{D}|\theta_i, M)^{\Delta\beta_j}} \tag{3.7}$$

When this is done, N_c Markov chains are initialised, each having the individual current samples θ_i as the seed sample. As such, N_c corresponds to the number of samples N. With probability $\hat{w}_k^j$, the k^{th} Markov chain (for $k = 1, \ldots, N$) is updated where a candidate sample θ_i^* is generated from a Normal proposal distribution $q(\theta_i^*|\theta_k)$ with mean θ_k and covariance matrix $\boldsymbol{\Sigma}^j$ [51]. Note that k is a dummy index denoting the index of the Markov chain selected via the weighted random sampling procedure and this is different from the index i which denotes the sample index.

The covariance matrix also serves as the tuning parameter of the MH sampler and is mathematically defined in Eq. (3.8):

$$\boldsymbol{\Sigma}^j = \gamma^2 \sum_{i=1}^{N} \hat{w}_i^j \cdot \left[\{\theta_i - \bar{\theta}^j\}^{\mathrm{T}} \times \{\theta_i - \bar{\theta}^j\}\right] \tag{3.8}$$

whereby

$$\bar{\theta}^j = \sum_{i=1}^{N} \theta_i \cdot \hat{w}_i^j \tag{3.9}$$

Here, $\bar{\theta}^j$ denotes the mean value of the sample set θ_i in the current iteration j, and γ is the scaling parameter of $\boldsymbol{\Sigma}^j$ to which a recommended value is 0.2 [51]. From there, θ_i^* is accepted or rejected using Algorithm 2. In the case where θ_i^* is accepted, θ_i^* is set as the current and terminal sample of the k^{th} Markov chain. Otherwise, θ_k remains as the current and terminal sample of the k^{th} Markov chain. This procedure of updating the samples is repeated N times. When done, the algorithm proceeds to recompute β_{j+1} and the the transitional distribution P^{j+1} for iteration $j = j + 1$. This entire process is repeated until when $\beta_j = 1$.

As seen in the description of the sampling procedure implemented by the TMCMC sampler, an important aspect is the determination of β_j and the transition step size $\Delta\beta_j$ (i.e. $\Delta\beta_j = \beta_j - \beta_{j-1}$) at each transitional step j. It has to be such that the transition from $P(\boldsymbol{D}|\boldsymbol{\theta}, M)^{\beta_{j-1}}$ to $P(\boldsymbol{D}|\boldsymbol{\theta}, M)^{\beta_j}$ is smooth and gradual. The magnitude of $\Delta\beta_j$ would have a direct impact on the acceptance rates of candidate samples generated via the MH sampling step. To demonstrate this, the same set-up used for Figure 3.5 is used. The value of $\Delta\beta_j$ is varied for which β_{j-1} is fixed at 0. From which, the value of β_j is varied and the corresponding value of acceptance rate will be obtained and shown in Figure 3.6. From Figure 3.6, it is observed that the value of acceptance rate drops from 1 when $\beta_j = 0$ to approximately 0.18 when $\beta_j = 1$. This is due to the large difference in the shape function between the prior and the posterior when β_j is large. Therefore, the majority of the candidate samples generated from the Uniform prior are rejected via the MH sampling procedure. On the other hand, when $\beta_j = 0$, the acceptance rate is 1 given that the samples are generated from the same distribution, leading to a 100 % acceptance rate. This illustrates the need to determine an optimal $\Delta\beta_j$.

To identify the optimal value of $\Delta\beta_j$, Ching and Chen (2007) suggested to maintain the coefficient of variation (COV) of the value set $P(\boldsymbol{D}|\theta_i, M)^{\Delta\beta_j}$ as close to 100 % as possible [51]. For a one-dimensional case, the COV of $P(\boldsymbol{D}|\theta_i, M)^{\Delta\beta_j}$ is defined as:

$$\mathrm{COV}(\beta_j) = \frac{\sigma(P(\boldsymbol{D}|\theta_i, M)^{\Delta\beta_j})}{\mu(P(\boldsymbol{D}|\theta_i, M)^{\Delta\beta_j})} \tag{3.10}$$

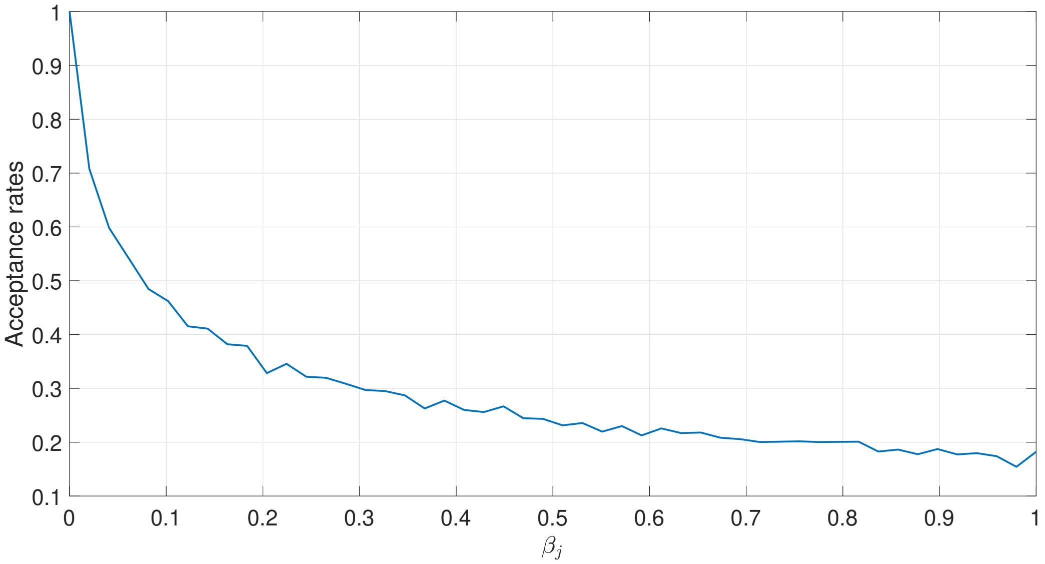


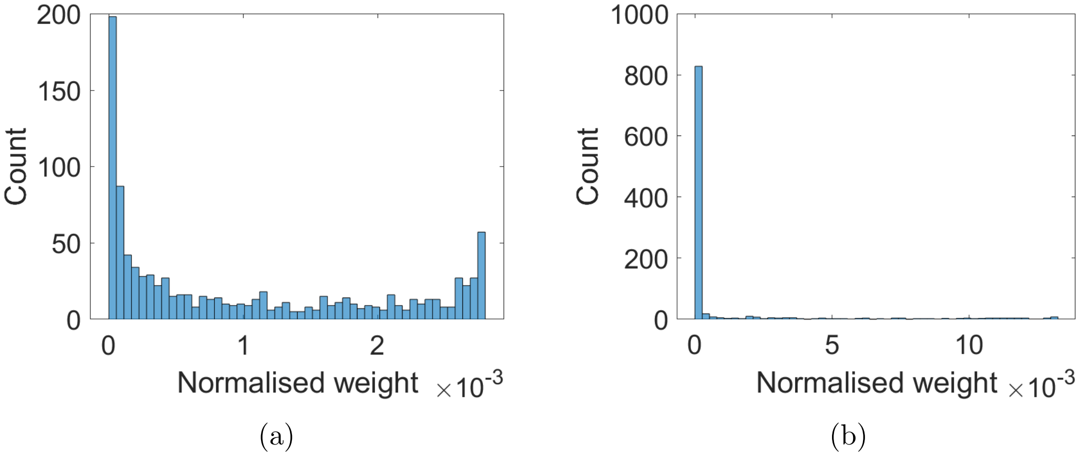
Figure 3.6: Plot of acceptance rates against the tempering parameter β_j based on the same set-up used for Figure 3.5.

whereby $\sigma(P(\boldsymbol{D}|\theta_i, M)^{\Delta\beta_j})$ and $\mu(P(\boldsymbol{D}|\theta_i, M)^{\Delta\beta_j})$ are the standard deviation and mean of the value set $P(\boldsymbol{D}|\theta_i, M)^{\Delta\beta_j}$, for $i = 1, ..., N$. Here, COV(β_j) is a function of β_j for a given known value of β_{j-1}. For a multi-dimensional case, $\sigma(P(\boldsymbol{D}|\boldsymbol{\theta}_i, M)^{\Delta\beta_j})$ and $\mu(P(\boldsymbol{D}|\boldsymbol{\theta}_i, M)^{\Delta\beta_j})$ are the standard deviation and mean of the value set $P(\boldsymbol{D}|\boldsymbol{\theta}_i, M)^{\Delta\beta_j}$ respectively whereby $P(\boldsymbol{D}|\boldsymbol{\theta}_i, M)$ is simply the likelihood evaluated at sample set $\boldsymbol{\theta}_i = (\theta_i^1, \theta_i^2, ..., \theta_i^{N_d-1}, \theta_i^{N_d})$. After obtaining COV$(\beta_j)$, β_j can then be determined analytically from β_{j-1} using the argument of the minimum of the absolute difference between COV(β_j) and 1 (i.e. 100 %) as shown in Eq. (3.11) [51, 169]:

$$\beta_j = \text{argmin}_{\beta_j} \{|\text{COV}(\beta_j) - 1|\} \tag{3.11}$$

Once β_j is calculated, the transition distribution P^j can then be determined using Eq. (3.6).

One notable advantage of using transitional distributions, with controlled transition step size, is that it helps to address the issue of degeneracy. Degeneracy occurs when only a few out of a total N samples have significant statistical weights associated with them. As an illustration, the same set-up which was used to produce Figure 3.5 is used. In this example, 2 values of transition step size from the Uniform prior are used: $\Delta\beta_j = 8.35 \times 10^{-3}$ (optimised step size); and $\Delta\beta_j = 0.204$ (larger step size). For each value of $\Delta\beta_j$, the distribution of the normalised weight across 1000 samples is obtained and presented in the form of histograms which are presented in Figure 3.7. With $\Delta\beta_j = 8.35 \times 10^{-3}$, Figure 3.7(a) shows a general uniform distribution of normalised weight values whereby every value has more or less the same number of samples having that associated weight. This is with exception to smaller weight values (near 0) where there is significantly higher counts of samples, approximately 198 out of 1000 (i.e. 19.8 % of samples), having such weight values. The reason for this is due these samples now lying in the region of lower probability defined by the transitional distribution. On the other hand, with $\Delta\beta_j = 0.204$, Figure 3.7(b) shows that the majority of the samples have very small weight values (near 0). In fact, from the histogram, approximately 828 out of 1000 samples (i.e. 82.8 % of samples) have such small values of associated weights. This illustrates degeneracy.



(a)

(b)

Figure 3.7: Normalised weight distribution when: (a) $\Delta\beta_j = 8.35 \times 10^{-3}$; and (b) $\Delta\beta_j = 0.204$. The transition step $\Delta\beta_j$ is with respect to the Uniform prior in Figure 3.5.

Based on the description above, it can be seen that the TMCMC sampling technique is able to generate N samples simultaneously per iteration, whereas the MCMC sampling technique could only compute one new proposal sample per iteration. In addition, the TMCMC sampling technique ensures that the samples in the j^{th} transition step are approximately distributed as per P^j, thereby making the need for burn-in unnecessary [51, 169].

3.2.2 Algorithmic description

The TMCMC sampler algorithm used to generate N samples from a one-dimensional posterior is summarised as follows [51]:

1. Set $j = 0$ and $\beta_j = 0$. Sample $\theta_i \sim P(\theta|M)$, for $i = 1, ...N$;
2. Set $j = j + 1$;
3. Compute β_j using Eq. (3.11);
4. While $\beta_j < 1$, compute P^j using Eq. (3.6);
5. Compute $\hat{w}_i^j$ using Eq. (3.7);
6. Generate N Markov chains each initiating from θ_i;
7. With probability $\hat{w}_k^j$, update the k^{th} Markov chain by first generating a candidate sample $\theta_i^* \sim q(\theta_i^*|\theta_k)$;
8. Compute the acceptance probability α using Eq. (3.3) and accept θ_i^* with probability α (see Algorithm 2);
9. If accepted, set θ_i^* as the current and terminal sample of the k^{th} Markov chain. Otherwise, set θ_k as the current and terminal sample of the k^{th} Markov chain;
10. Repeat Steps (7) to (9) N times;
11. Repeat Steps (2) to (10) until $\beta_j = 1$.

To sample from a multi-variate posterior, a component-wise approach is adopted in the chapter for the TMCMC sampler [170]. Such approach is chosen due to the simplicity in its implementation. The algorithmic description of the TMCMC sampler in sampling from a general N_d-dimensional posterior is presented in Algorithm 3.

3.2.3 Applications

Currently, the TMCMC sampler has already been implemented in numerous applications to which examples include: 1) characterising the statistical uncertainties of the spatial variability parameters which are based upon the Cone Penetration Test [171]; 2) studying the multi-modality feature of the Bouc–Wen–Baber–Noori model of hysteresis [172]; 3) performing model updating and analyse the uncertainty associated with the creep behavior of soft soil [173]; 4) performing reliability-based optimization in linear structure designs subjected to random excitations [174]; 5) analysing the geometrical uncertainty of a metal frame [175];

Algorithm 3 Component-wise TMCMC algorithm (N_d-dimensional case)

```
procedure (Generate $N$ samples from a general $N_d$-dimensional posterior)
    Set $j = 0$ and $\beta_j = 0$                                    ▷ Initialise
    for $i = 1 : N$ do
        Draw initial sample set: $\boldsymbol{\theta}_i \sim P(\boldsymbol{\theta}|M)$
    end for
    while $\beta_j < 1$ do                                           ▷ Main sampling loop
        Set $j = j + 1$
        Compute $\Delta\beta_j$ using Eq. (3.11)
        Compute $P^j$ using Eq. (3.6)
        for $i = 1 : N$ do                                           ▷ Updating step
            Select the $k^{th}$ Markov chain $\sim \hat{w}_k^j$
            for $d = 1 : N_d$ do                                     ▷ Update $d^{th}$ component
                Draw candidate sample: $\theta_i^{d*} \sim q(\theta_i^*|\theta_k^d)$
                Accept/Reject $\theta_i^{d*}$ using Algorithm 2 with 1 iteration
            end for
        end for
    end while
end procedure
```

6) providing a probabilistic hierarchical Bayesian framework for time-domain model updating [176]; 7) performing Bayesian inference for the identification of local structural properties of layered composites [78]; 8) performing cracks identification on beams through Bayesian approach [79]; 9) performing model parameter updating for piezoelectric energy harvesters [95]; and 10) performing inverse uncertainty quantification with limited experimental data [74].

3.3 Sequential Monte Carlo

3.3.1 Conceptual introduction

The Sequential Monte Carlo (SMC) sampler is a sampling approach that is based on the SMC methods (or Particle filter) and developed to sample from a sequence of probability distributions in an iterative (i.e. sequential) manner [52]. In the context of Bayesian model updating, the SMC sampler becomes favourable for target tracking [52, 177, 178, 179, 180] which makes it practical and efficient in the following 3 cases:

1) when a large data set is available and it may become computationally expensive to perform

the Bayesian model updating procedure with the entire data set, thereby bringing the need to divide the data set into smaller batches [181];

2) when data is being collected at different time intervals (i.e. on-line learning) and it becomes inefficient to use sampling techniques such as the MCMC samplers which requires a complete data set to be available in order to sample from the posterior [52, 182]; and

3) when the posterior is functionally complex (i.e. multi-modal or highly-skewed) and it becomes difficult to sample from such posteriors directly [183].

The choice of probability distribution sequence is an important aspect for consideration when implementing the SMC sampler to address each case.

Given n sequence of data sets such that $\boldsymbol{D}^{1:n} = \{\boldsymbol{D}^1, \dots, \boldsymbol{D}^n\}$, there are 2 types of probability distribution sequence P^j that can be used. The first type takes the following form [181]:

$$P^j \propto P(\boldsymbol{\theta}|\boldsymbol{D}^{1:j}, M) \cdot P(\boldsymbol{\theta}|M) \tag{3.12}$$

whereby

$$P(\boldsymbol{\theta}|\boldsymbol{D}^{1:j}, M) = \prod_{s=1}^{j} P(\boldsymbol{\theta}|\boldsymbol{D}^{s}, M) \tag{3.13}$$

and $j \leq n$ is the sampling iteration number. This set-up would be applicable in addressing cases (1) and (2) under the assumption that each sequence of data set is independent from one another [82]. In addition, such method of constructing P^j would be optimal when there is a need to know the posterior distribution and the estimate of $\boldsymbol{\theta}$ at any given iteration j. In fact, the sequential Bayesian inference framework for the on-line inference on $\boldsymbol{\theta}$ can be described by Eq. (3.12) [181]. The second type takes the following form [52, 141, 184]:

$$P^j = P(\boldsymbol{\theta}|\boldsymbol{D}^{1:n}, M)^{\beta_j} \cdot P(\boldsymbol{\theta}|M)^{1-\beta_j} \tag{3.14}$$

where $\beta_0 = 0 < \beta_1 < ... < \beta_{j_{end}-1} < \beta_{j_{end}} = 1$, and j_{end} denotes the last iteration number. Given that the posterior $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$ is defined as per Eq. (2.8), the expression in Eq. (3.14) can be simplified to the form as shown in Eq. (3.6). This set-up would be applicable in addressing cases (1) and (3). In fact, such set-up is adopted by the TMCMC sampler as presented in Section 3.2. Hence, the TMCMC sampler can be seen as a particular variant of the general SMC samplers [169] despite the TMCMC sampler being developed independently from the SMC sampler and that the former is based on a MCMC technique [51]. For both probability distribution sequences defined in Eq. (3.12) and Eq. (3.14), P^0 is simply the prior $P(\boldsymbol{\theta}|M)$.

Another key aspect in the SMC sampling procedure is the assignment and updating of weights on the individual sample. This is done in an iterative manner through Importance sampling [185, 186, 135]. In sampling from the first sequence of distribution at iteration $j = 1$ (i.e. P^1), the un-normalised weight on each sample $\boldsymbol{\theta}_i$ is computed following [52]:

$$w_i^1 = \frac{P^1(\boldsymbol{\theta}_i)}{P^0(\boldsymbol{\theta}_i)} \tag{3.15}$$

For subsequent iterations $j \geq 2$, the un-normalised weight on each sample $\boldsymbol{\theta}_i$ is updated recursively following [52]:

$$w_i^j = w_i^{j-1} \cdot \frac{P^j(\boldsymbol{\theta}_i^*)}{P^{j-1}(\boldsymbol{\theta}_i)} \cdot \frac{q_L(\boldsymbol{\theta}_i|\boldsymbol{\theta}_i^*)}{q_K(\boldsymbol{\theta}_i^*|\boldsymbol{\theta}_i)} \tag{3.16}$$

where $q_K(\boldsymbol{\theta}_i^*|\boldsymbol{\theta}_i)$ denotes the Forward Markov kernel, and $q_L(\boldsymbol{\theta}_i|\boldsymbol{\theta}_i^*)$ denotes the Backward Markov kernel.

The Forward Markov kernel $q_K(\boldsymbol{\theta}_i^*|\boldsymbol{\theta}_i)$ serves to move the samples obtained from P^{j-1} and update them according to the current distribution P^j. This is done through the following choice of move kernels which include [52]: 1) Independent proposals in the form of standard distributions such as a Normal distribution; 2) Local random-walk moves; 3) MCMC moves; and 4) Approximate Gibbs moves. For the dissertation, the MCMC move in the form of the MH algorithm is chosen as the Forward Markov kernel. The Backward Markov kernel $q_L(\boldsymbol{\theta}_i|\boldsymbol{\theta}_i^*)$ serves to approximate the distribution of P^{j-1} based on the samples from P^j. Optimally, the choice of $q_L(\boldsymbol{\theta}_i|\boldsymbol{\theta}_i^*)$ is one that minimises the variance of the sample estimates on $\boldsymbol{\theta}$ at any given iteration j and this is achieved when the kernel takes the form [52]:

$$q_L^{opt}(\boldsymbol{\theta}_i|\boldsymbol{\theta}_i^*) = \frac{P^{j-1}(\boldsymbol{\theta}_i) \cdot q_K(\boldsymbol{\theta}_i^*|\boldsymbol{\theta}_i)}{P^j(\boldsymbol{\theta}_i^*)} \tag{3.17}$$

However, it is generally difficult to implement $q_L^{opt}(\boldsymbol{\theta}_i|\boldsymbol{\theta}_i^*)$ in practice given that it requires the marginal distributions of $P^{j-1}(\boldsymbol{\theta}_i)$ and $P^j(\boldsymbol{\theta}_i^*)$ to be known exactly and expressed in closed-form. Such exact knowledge on the marginal distributions is never available in general [52]. For this reason, a sub-optimal implementation of $q_L^{opt}(\boldsymbol{\theta}_i|\boldsymbol{\theta}_i^*)$ is used in the dissertation. Given that the MH move kernel is chosen as the choice of $q_K(\boldsymbol{\theta}_i^*|\boldsymbol{\theta}_i)$ in the implementation of the

SMC sampler in the dissertation, an approximated $q_L(\boldsymbol{\theta}_i|\boldsymbol{\theta}_i^*)$ is used which takes the form [52]:

$$q_L(\boldsymbol{\theta}_i|\boldsymbol{\theta}_i^*) = \frac{P^j(\boldsymbol{\theta}_i) \cdot q_K(\boldsymbol{\theta}_i^*|\boldsymbol{\theta}_i)}{P^j(\boldsymbol{\theta}_i^*)} \tag{3.18}$$

Substituting Eq. (3.18) into Eq. (3.16), the un-normalised weight w_i^j at iteration $j \geq 2$ can therefore be simplified as:

$$w_i^j = w_i^{j-1} \cdot \frac{P^j(\boldsymbol{\theta}_i)}{P^{j-1}(\boldsymbol{\theta}_i)} \tag{3.19}$$

Note that the above settings describes a specific implementation of the SMC sampler which will be used in the dissertation and is not the general implementation of the sampler. To emphasise this, the SMC sampler will subsequently be referred to as the SMC with MH kernel (SMC-MH) sampler.

In addressing the case studies presented in the chapter and the experimental example presented in Chapter 8 of the dissertation, the probability distribution sequence defined in Eq. (3.12) will be used in the implementation of the SMC-MH sampler to perform sequential Bayesian inference on $\boldsymbol{\theta}$. To provide an understanding of the SMC-MH sampler, an explanation to the procedure behind sampling from a one-dimensional posterior will first be provided. At iteration $j = 0$, N samples are generated from the prior via direct random sampling using the Monte Carlo method. At iteration $j = 1$, each sample θ_i (for $i = 1, \dots, N$) is assigned a statistical (or importance) weight w_i^j using Eq. (3.15). In the case of P^j defined in Eq. (3.12) being used as the distribution sequence, w_i^j is simply the likelihood function obtained accounting for the first sequence of data (i.e. $P(\boldsymbol{D}^1|\theta_i, M)$). From which the normalised importance weight $\hat{w}_i^j$ is obtained using Eq. (3.20):

$$\hat{w}_i^j = \frac{P(\boldsymbol{D}^j|\theta_i, M)}{\sum_{i=1}^{N} P(\boldsymbol{D}^j|\theta_i, M)} \tag{3.20}$$

After which, N samples of θ_i are resampled according to $\hat{w}_i^j$ using the weighted resampling approach (with replacement). An illustration to the weighted resampling procedure is provided in Figure 3.8. In doing so, samples with higher weights are being resampled more often, eventually discarding samples with relatively “insignificant” weights. Such procedure, however, only helps to artificially “conceal” impoverishment by ensuring that unique samples with high associated weights are being duplicated to a higher extent which introduces high correlations between the samples, and does not contribute in the exploration of the sample space [181]. In order to generate more unique samples and update those samples according to the current distribution P^j, N_c Markov chains are initialised from each of the resampled θ_i thereby giving rise to N of such Markov chains. From there, the MH sampler is implemented to generate 1 sample from each Markov chain [50]. The candidate samples are generated from a Normal proposal distribution $q(\theta_i^*|\theta_i)$ with mean $\bar{\theta}$ and covariance matrix $\mathbf{\Sigma}^j$ which are computed using Eq. (3.9) and Eq. (3.8) respectively. In computing $\mathbf{\Sigma}^j$, the scaling parameter γ is set as 1 as suggested in [181]. The candidate sample θ_i^* for a given i^{th} Markov chain is then accepted or rejected using Algorithm 2. This procedure is repeated for all N Markov chains. When this is done, the un-normalised weights of the individual resampled θ_i are reset to $w_i^j = 1$ given that the above procedure of updating the samples ensures that the samples are distributed according to the moments of P^j. From there, the algorithm proceeds to iteration $j = j + 1$, re-computes the normalised weights $\hat{w}_i^j$ using Eq. (3.20) (i.e. since $w_i^{j-1} = 1$), and the resampling and sample updating steps are undertaken. This sampling procedure is repeated for each new data sequence $\boldsymbol{D}^j$ until the last data sequence is obtained at iteration $j = j_{end}$.

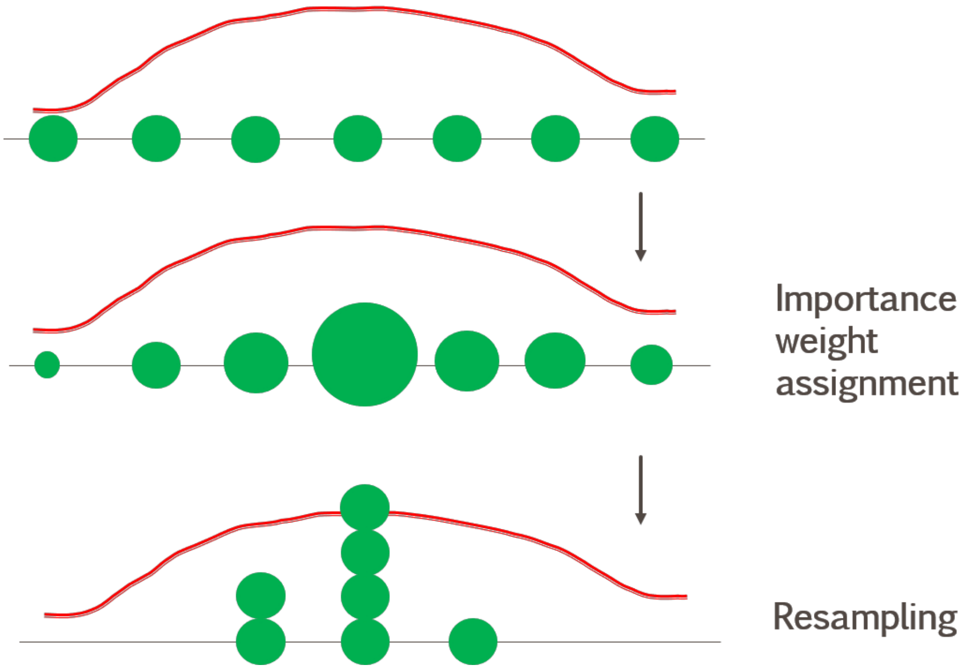


Figure 3.8: Illustration as to how the resampling procedure is done according to the statistical weights of the samples in green. The red curve here represents the statistical weight function $\hat{w}_i^j$.

There are 2 notable similarities between the TMCMC and the general SMC sampling techniques as highlighted in the previous paragraph of this section: 1) both the TMCMC and SMC sampling algorithms are capable sampling from a sequence of distributions in a sequential manner as highlighted in Section 3.3.1; and 2) both sampling techniques are able to generate all N samples within each iteration. There are, however, 4 main differences between the 2 techniques [51, 169, 52, 181]: 1) the TMCMC sampler is mainly implemented in an off-line Bayesian inference set-up whereas the SMC sampler can be implemented in off-line and/or on-line Bayesian inference set-ups (although the SMC sampler is optimised for the latter); 2) the TMCMC sampler is mainly implemented to infer time-invariant parameter(s) whereas the SMC sampler can be implemented to infer time-invariant and/or time-varying parameter(s); 3) in updating the samples, the TMCMC sampling algorithm ensures that samples with higher $\hat{w}_i^j$ at any given iteration j are assembled into longer Markov chains than those with lower $\hat{w}_i^j$ [187], whereas the SMC sampling algorithm would initiate more Markov chains from samples with higher $\hat{w}_i^j$ leading to a higher concentration of Markov chains in regions of the posterior with higher probabilities; and 4) due to the different sample updating strategies as highlighted in the previous point, the SMC sampling algorithm is paralleliseable [182, 188] whereas the TMCMC sampling algorithm is difficult to parallelise. It is, however, still possible for the TMCMC sampling algorithm to be parallelised although such capability is constrained by the diversity of the resampled sample set.

3.3.2 Algorithmic description

The SMC-MH sampler algorithm used to generate N samples from a one-dimensional posterior is summarised as follows [181]:

1. At iteration $j = 0$, sample $\theta_i \sim P(\theta|M)$ for $i = 1, ..., N$;

2. Set $j = j + 1$;

3. Calculate $\hat{w}_i^j$ using Eq. (3.20);

4. Resample $\theta_i \sim \hat{w}_i^j$, for $i = 1, \dots, N$;

5. Generate N Markov chains each initiating from the resampled θ_i;

6. For each i^{th} chain, generate candidate sample $\theta_i^* \sim q(\theta_i^*|\theta_i)$;

7. Compute the acceptance probability α using Eq. (3.3) and accept θ_i^* with probability α (see Algorithm 2);

8. Reset the un-normalised weights $w_i^j = 1$, for $i = 1, \ldots, N$;

9. Repeat Steps (2) to (8) until the last set of observations $\boldsymbol{D}$ is obtained at $j = j_{end}$.

To sample from a multi-variate posterior, a component-wise approach is adopted in the chapter for the SMC-MH sampler. Such approach is chosen due to the simplicity in its implementation. The algorithmic description of the SMC-MH sampler in sampling from a general N_d-dimensional posterior is presented in Algorithm 4.

Algorithm 4 Component-wise SMC-MH sampler algorithm (N_d-dimensional case)

```
procedure (Generate $N$ samples from a general $N_d$-dimensional posterior)
    Set $j = 0$                                                    ▷ Initialise
    for $i = 1 : N$ do
        Draw initial sample set: $\boldsymbol{\theta}_i \sim P(\boldsymbol{\theta}|M)$
    end for
    while $j < j_{end}$ do                                         ▷ Main sampling loop
        Set $j = j + 1$
        Compute $\hat{w}_i^j$ using Eq. (3.20)
        for $i = 1 : N$ do                                         ▷ Updating step
            Resample: $\boldsymbol{\theta}_i^j \sim \hat{w}_i^j$
            for $d = 1 : N_d$ do                                   ▷ Update $d^{th}$ component
                Draw candidate sample: $\theta_i^{d*} \sim q(\theta_i^*|\theta_i^d)$
                Accept/Reject $\theta_i^{d*}$ using Algorithm 2 with 1 iteration
            end for
        end for
        for $i = 1 : N$ do                                         ▷ Reset the weights
            Set $w_i^j = 1$
        end for
    end while
end procedure
```

3.3.3 Applications

Currently, the SMC sampler has already been implemented in numerous applications to which examples include: 1) analysing and quantifying the uncertainty of the measured data from probabilistic nonlinear state-space models of an engineering dynamical system [189]; 2) performing uncertainty reduction in prognostics [190]; 3) estimating parameters of dynamical engineering systems from big data [82]; 4) performing Bayesian learning of state-space models with highly informative observations [191]; 5) extracting bearing fault features via Bayesian approach [192]; 6) simultaneous output and model parameter estimation for nonlinear engineering systems under low measurement rate constraints [193]; 7) identifying stiffness parameters of a building structure for damage detection using response frequency data [194]; 8) real time residual lifetime prediction on a stiffened panel subject to fatigue crack growth [195]; and 9) real time prognosis of crack growth evolution on a critical engineering component [196].

3.4 Case study 1: Spring-mass system

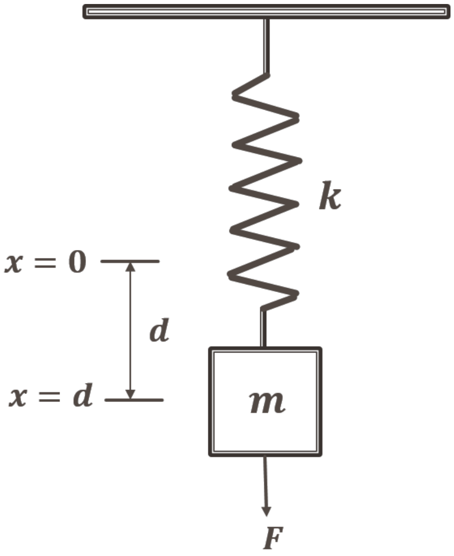


Figure 3.9: Schematic diagram of the simple spring-mass system.

Figure 3.9 illustrates a spring-mass system consisting of a mass m attached to a spring k, subject to a static force F. The initial position of the mass is $x = 0$. When F is applied to the mass, the mass will move to a new position $x = d$. It is well known that for this type of problem F and d are related by Hooke's Law so that:

$$F = -k \times d \tag{3.21}$$

In this application, it is assumed that k has a fixed value of 263 N/m which is uncertain. However the measurements of d are affected by measurement "noise" such that:

$$d_{measured} = d + \epsilon_1 \tag{3.22}$$

In addition, the measurements of F are also affected by measurement "noise" such that:

$$F_{measured} = F + \epsilon_2 \tag{3.23}$$

The parameters ϵ_1 and ϵ_2 are assumed to be independent random variables following a Normal distribution with means 0.0 N and standard deviations 0.003 m and 1.0 N respectively. Overall, the total effect of the "noise" on the data obtained for $F_{measured}$ is contributed by "noise" in the measurements of both quantities d and F as seen in Eq. (3.22) and (3.23) as well as in reality. However, only the contribution of measurement "noise" from $F_{measured}$ will be considered. This is due to k being the inferred parameter of the model defined in Eq. (3.21) which is used to predict the quantity F. For this problem, 15 independent realisations of the $F_{measured}$ - $d_{measured}$ pair are obtained. The $F_{measured}$ - $d_{measured}$ data obtained are presented in the form of a scatterplot shown in Figure 3.10 while its numerical values are presented in a table as shown in Table 3.3.

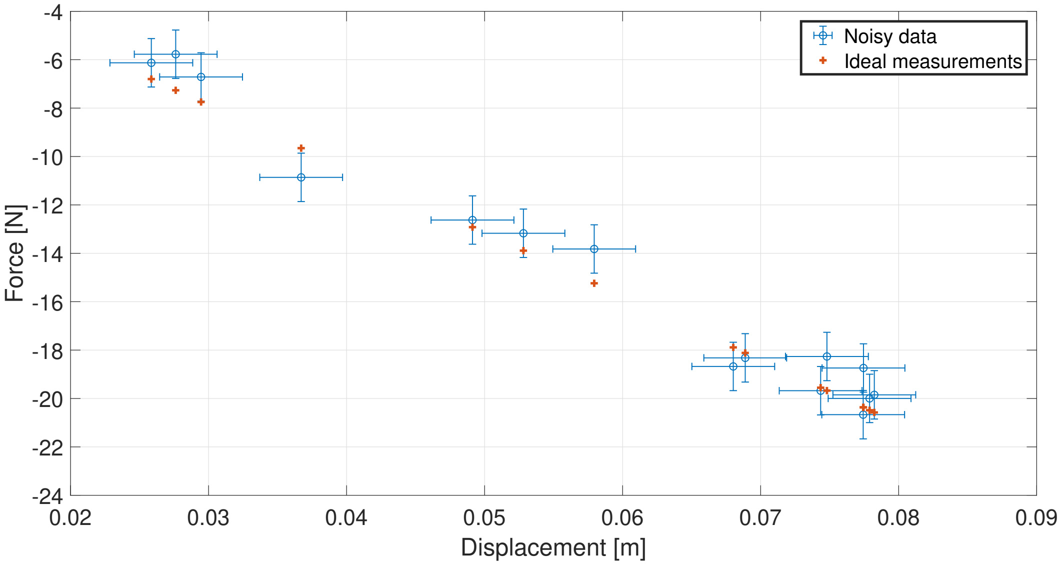


Figure 3.10: scatterplot of the 15 simulated "noisy" data of Force against the respective values of Displacements.

Measurement no.	d $[m]$	$F_{measured}$ $[N]$	Ideal measurement $[N]$
1	0.0259	-6.13	-6.80
2	0.0276	-5.77	-7.26
3	0.0295	-6.71	-7.75
4	0.0367	-10.86	-9.65
5	0.0491	-12.63	-12.92
6	0.0528	-13.17	-13.89
7	0.0579	-13.82	-15.24
8	0.0680	-18.68	-17.89
9	0.0688	-18.32	-18.12
10	0.0743	-19.68	-19.55
11	0.0748	-18.26	-19.67
12	0.0774	-20.67	-20.36
13	0.0775	-18.74	-20.37
14	0.0779	-20.00	-20.49
15	0.0782	-19.85	-20.58

Table 3.3: Numerical values of the data illustrated in Figure 3.10.

3.4.1 Linear Least-squares Method

One direct way to solve for k analytically would be via the method of Linear Least-squares minimization [30]. The equations to the Linear Least-squares method is as follows:

$$(\mathbf{x}^T\mathbf{x})k = \mathbf{x}^T F \tag{3.24}$$

whereby $\mathbf{x}$ is the design matrix, which in this case would be the vector of the displacement values d, $\mathbf{x}^T$ is the transpose of the design matrix, and F is the vector of the measured values of the force acting on the spring. As such, k can be solved by re-expressing Eq. (3.24) into the following form:

$$k = (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T F \tag{3.25}$$

Using the data values shown in Table 3.3 as well as the left matrix divide operation on MATLAB, the Linear Least-squares solution to k is 255.87 N/m with a percentage discrepancy of -2.71 % from its true value. The updated linear model is illustrated in Figure 3.11.

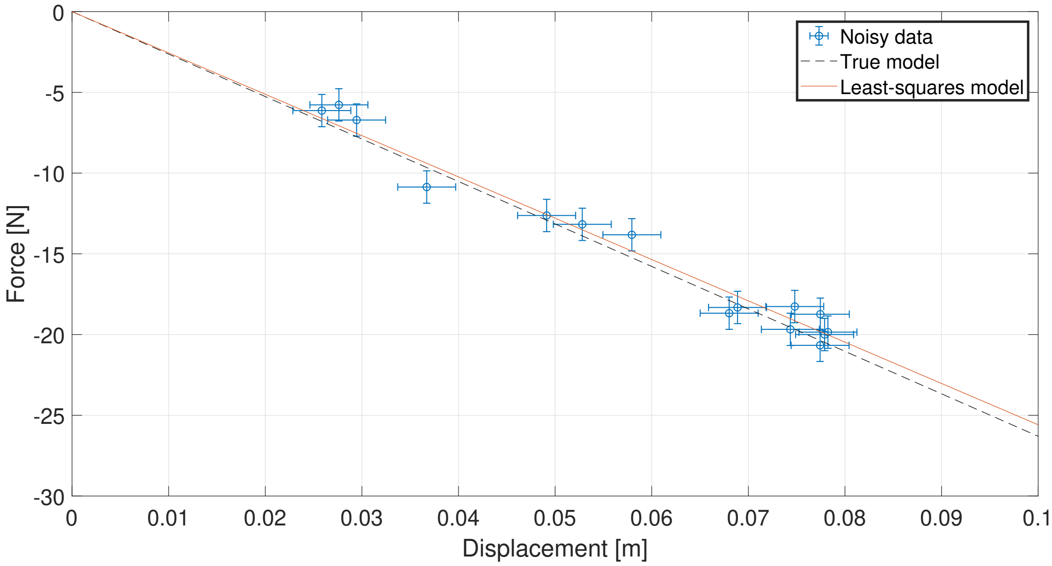


Figure 3.11: The updated linear model via the linearisation method as illustrated by the red line with the "noisy" data represented by the blue circles.

3.4.2 Bayesian model updating

For this problem, the a-priori knowledge of k is based on the initial hypothesis that k can range between 200 N/m and 700 N/m. As such, the prior distribution of k, $P(k|M)$ (i.e. M is represented by Eq. (3.21)), can be modelled after a Uniform distribution whose lower-bound and upper-bound values are 200 N/m and 700 N/m respectively. The likelihood function is modelled using a Normal distribution with the standard deviation equal to that of the noise, ϵ. Thus, the likelihood function is expressed as follows:

$$P(F_{measured,1:15}|k, M) \propto \exp\left[-\frac{1}{2 \cdot {\sigma_{\epsilon}}^2} \sum_{n=1}^{15} (F_{measured,n} - M(k))^2\right] \tag{3.26}$$

where $\sigma_{\epsilon} = 1.0$ N/m. The justification behind the choice of the likelihood function and its associated standard deviation is as presented in Chapter 2. The above set-up for the prior and likelihood function yields an analytical solution to the resulting posterior which is expressed as follows:

$$P(k|F_{measured,1:15}, M) \propto \exp\left[-\frac{(k - \mu_k)^2}{2 \cdot {\sigma_k}^2}\right] \tag{3.27}$$

whereby

$$\mu_k = \frac{\sum_{n=1}^{15} F_{measured,n} \cdot d_n}{\sum_{n=1}^{15} {d_n}^2} \tag{3.28}$$

$$\sigma_k = \frac{\sigma_\epsilon}{\sqrt{\sum_{n=1}^{15} {d_n}^2}} \tag{3.29}$$

From Eq. (3.28) and Eq. (3.29), this yields $\mu_k = 255.87\ N/m$ and $\sigma_k = 4.19\ N/m$.

In this problem, the sampling performances of the MCMC, TMCMC, and the SMC-MH samplers are compared on the basis of: 1) the time elapsed in sampling from the posterior; and 2) their respective Monte Carlo standard error σ_{SE} on the estimation of k which is computed from Eq. (3.4) using batch size $n_b = 100$.

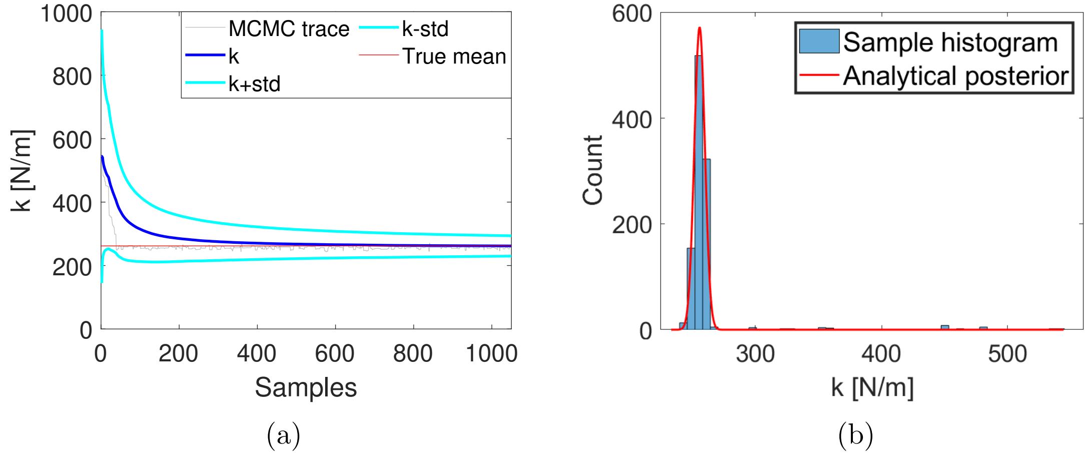
3.4.3 MCMC sampler results

Before the main sampling procedure is conducted, a calibration experiment is performed so as to determine the value of the tuning parameter whereby the MCMC sampler is able to achieve an acceptance rate close to the optimum value of 0.234. To do this, a fixed sample size of $N = 10000$ is obtained from the posterior, with 0 burn-in, using 5 different values of the tuning parameter: 0.10 N/m, 1.50 N/m, 22.50 N/m, 80.00 N/m, and 100.00 N/m. The numerical results are summarized in Table 3.4. Based on the results, the optimum value of tuning parameter is determined to be 22.50 N/m.

Tuning parameter $[N/m]$	Acceptance level	Time $[s]$	$N_{burn-in}$
0.10	0.906	1.80	Undetermined
1.50	0.844	1.78	1250
22.50	0.232	1.76	65
80.00	0.067	1.72	90
100.00	0.051	1.79	80

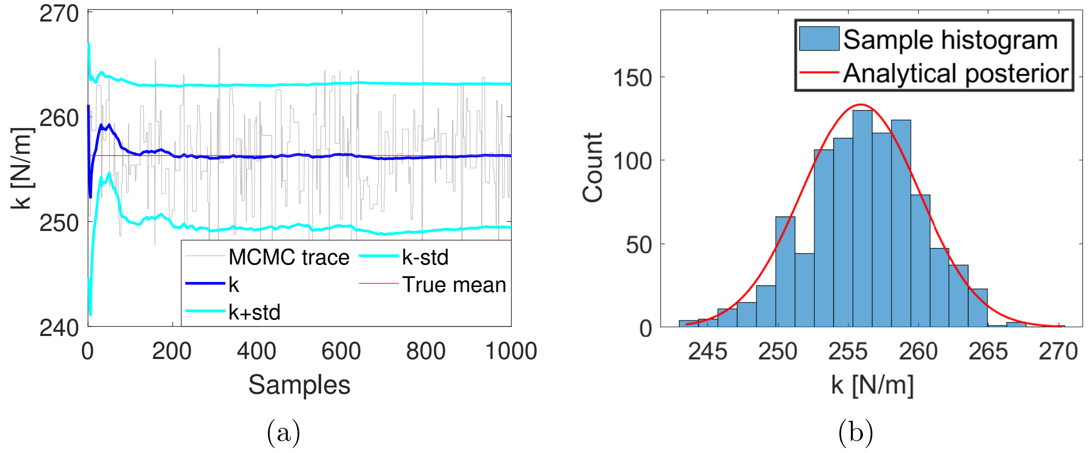
Table 3.4: Summary of results from varying the tuning parameter values while keeping the sample size fixed at $N = 10000$.

The the main sampling procedure is then conducted with a sample size of $N = 1050$. This value of sample size is chosen as it ensures sufficient convergence of the sample estimate of k by the MCMC sampler. The chain is initiated at 544.33 N/m and sampling is first done with 0 burn-in. The resulting trace plot and histogram of the sample values of k is shown in Figure 3.12. Based on the trace plot in Figure 3.12, the burn-in length is determined to be approximately 50. Figure 3.13 illustrates the resulting trace plot and histogram of the sample values of k after discarding the first 50 samples. The sampler took 0.31 seconds of computation time and yielded an estimated mean value of 256.29 N/m for k with a standard error σ_{SE} of 0.21 N/m.



(a) (b)

Figure 3.12: The resulting sample trace plot (a) and the histogram (b) obtained using the MCMC sampler with sample size $N = 1050$ with $N_{burn-in} = 0$. The red line in the sample trace plot denotes the true sample mean value.

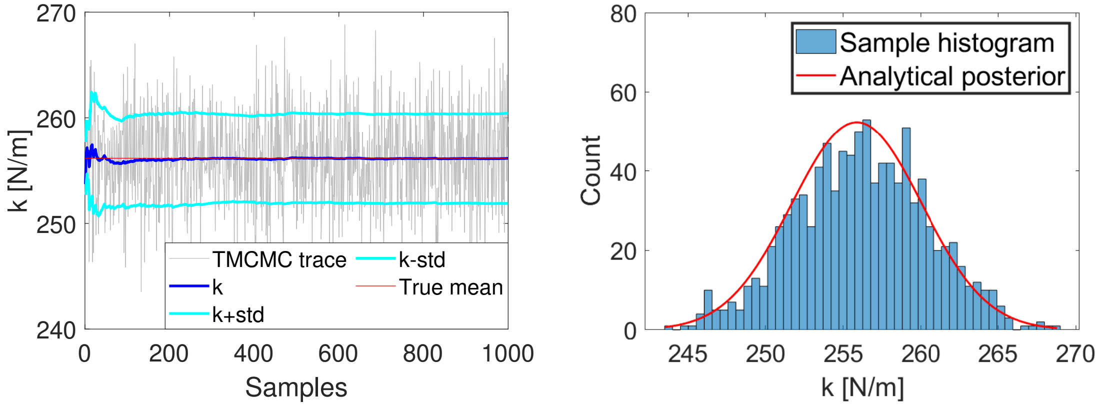


(a) (b)

Figure 3.13: The resulting sample trace plot (a) and the histogram (b) after discarding the first $N_{burn-in} = 50$ samples. The red line in the sample trace plot denotes the true sample mean value.

3.4.4 TMCMC sampler results

For the TMCMC sampler, a sample size of $N = 1000$ samples was generated from the posterior to ensure sufficient convergence of the sample estimate of k. The sampler took 7.68 seconds of computation time over 4 iterations and yielded an estimated mean value of 256.17 N/m for k with a standard error σ_{SE} of 0.10 N/m. The resulting trace plot and histogram are presented in Figure 3.14.



(a) (b)

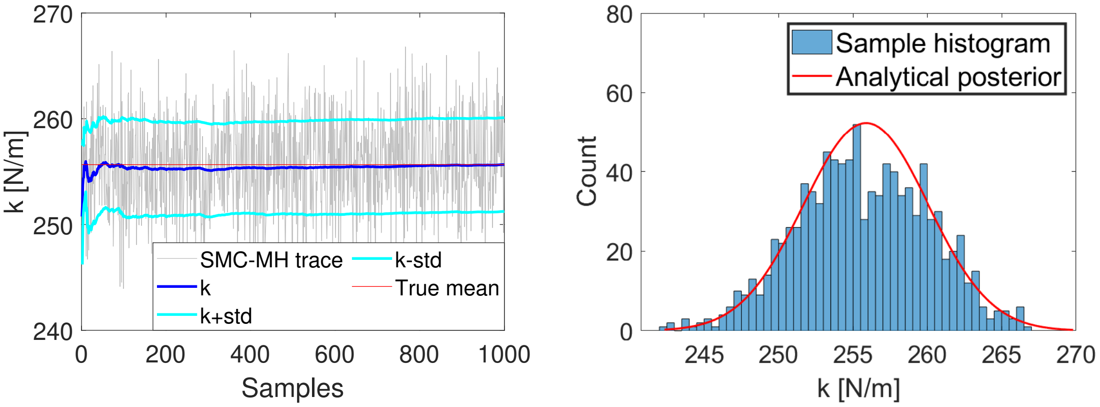
Figure 3.14: The resulting sample trace plot (a) and the histogram (b) obtained using the TMCMC sampler with sample size $N = 1000$. The red line in the sample trace plot denotes the true sample mean value.

3.4.5 SMC-MH sampler results

For the SMC-MH sampler, a sample size of $N = 1000$ samples was generated from the posterior to ensure sufficient convergence of the sample estimate of k. In sampling from the final posterior, 2 approaches are employed: 1) by batching the measurement data into 15 distinct batches and sampling from the posterior in a sequential manner as described in Section 3.3.1; and 2) by sampling directly from the final posterior.

For the first approach, the sampler took 200.36 seconds of computation time over 15 iterations and yielded an estimated mean value of 255.66 N/m for k with a standard error σ_{SE} of 0.15 N/m. The resulting sample trace plot and histogram are presented in Figure 3.15.

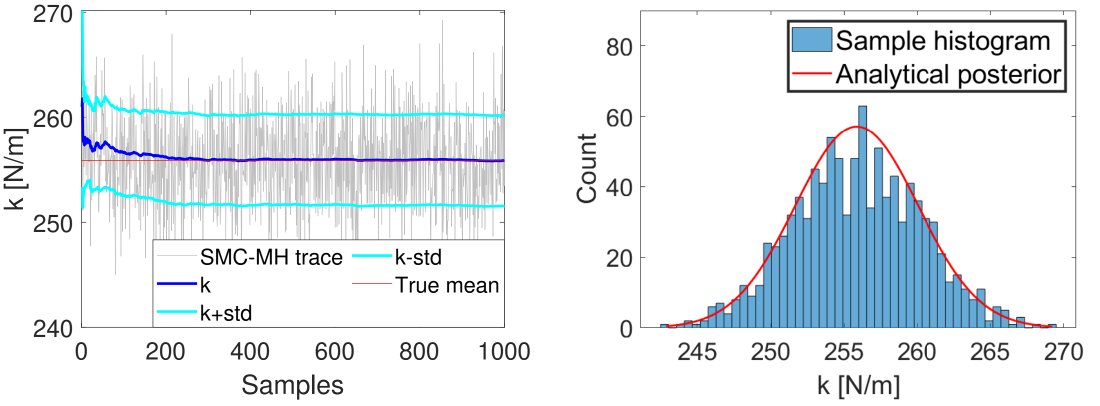
For the second approach, the sampler took 2.41 seconds of computation time over 1 iteration and yielded an estimated mean value of 255.88 N/m for k with a standard error



(a) (b)

Figure 3.15: The resulting sample trace plot (a) and the histogram (b) obtained using the SMC-MH sampler in sampling from the posterior via a sequential approach with sample size $N = 1000$. The red line in the sample trace plot denotes the true sample mean value.

σ_{SE} of 0.13 N/m. The resulting sample trace plot and histogram are presented in Figure 3.16.



(a) (b)

Figure 3.16: The resulting sample trace plot (a) and the histogram (b) obtained using the SMC-MH sampler in sampling directly from the posterior with sample size $N = 1000$. The red line in the sample trace plot denotes the true sample mean value.

Sampler	N	E[k] [N/m]	σ_{SE} [N/m]	Time [s]	Iterations
MCMC	1000	256.29	0.21	0.31	1050
TMCMC	1000	256.17	0.10	7.68	4
SMC-MH (batched)	1000	255.66	0.15	200.36	15
SMC-MH	1000	255.88	0.13	2.41	1

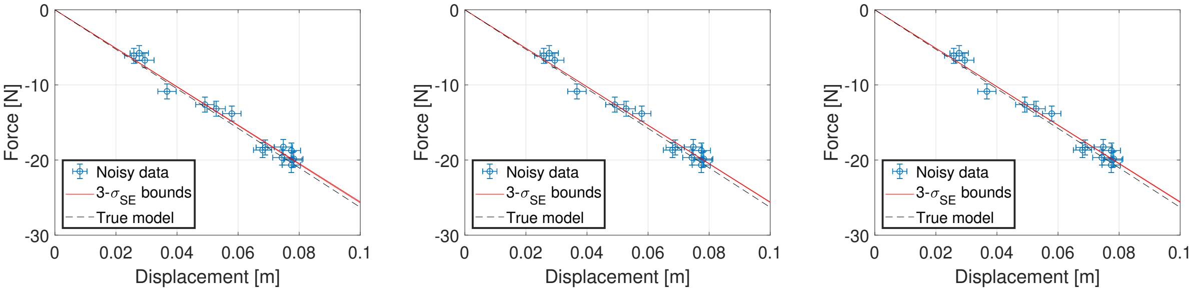
Table 3.5: Summary of the numerical results of the estimation of k by the respective samplers. Reference solution: $k = 263$ N/m, and $\sigma_{SE} = 0.13$ N/m (i.e. see Eq.(3.4)).

3.4.6 Discussions

The overall results of the sampling estimates of k for each sampler are summarized in Table 3.5 and the resulting Bayesian model update by each of the sampler are also presented in Figure 3.17. Based on the results, it is observed that the TMCMC and the SMC-MH sampling (i.e. the batched approach) algorithms implemented in the dissertation are significantly slower than that of the MCMC sampling algorithm implemented in the dissertation given the same number of samples obtained from the posterior. This is due to the fact that both the TMCMC and the SMC-MH sampling algorithms generate N samples from their respective transitional distributions at each iteration which, in turn, increases the number of model evaluations and the overall sampling time. In addition, it can be seen that there is no significant difference in the statistics of the estimation of k (i.e. the mean and standard error) by the SMC-MH sampler between either approaches as described in Section 3.4.5. As such, for subsequent case studies presented in the chapter, the approach of sampling directly from the final posterior (in 1 iteration) will be implemented for the SMC-MH sampler. Furthermore, it can be seen from the results that the estimate of k obtained by the TMCMC sampler has the lowest standard error while the MCMC sampler took the shortest time in sampling from the posterior.

3.5 Case study 2: 2D Bi-modal posterior

In this case study, the performance of the MCMC, TMCMC, and SMC-MH sampling techniques will be analysed in estimating the parameters of interest from a 2-dimensional, bi-modal posterior distribution. This will be done for a 2 × 2 square matrix, $\boldsymbol{H}$, which takes on the following form:



(a) MCMC model update.

(b) TMCMC model update

(c) SMC-MH model update

Figure 3.17: Results of the model updating for the respective samplers. The red lines denote the 3-σ_{SE} bounds.

$$\boldsymbol{H} = \begin{bmatrix} \theta_1 + \theta_2 & -\theta_2 \\ -\theta_2 & \theta_2 \end{bmatrix}$$

An example of a matrix taking on such form, in the context of engineering problems, would be the Stiffness matrix used to describe the configuration of a tuned mass damper system [197]. In this problem, θ_1 and θ_2 are the matrix elements which are assumed to have the following fixed values: $\{\theta_1, \theta_2\} = \{0.5, 1.5\}$.

In a physical context, $\boldsymbol{H}$ represents a physical system whose eigenvalues, denoted as λ_1 and λ_2, represent the possible observations that can be made from a given system. Readers of this article are assumed to be familiarized with the derivation of the eigenvalues. The actual observations λ_i^{noisy} are, however, corrupted with their respective "noise", ϵ_i, for $i = 1, 2$, such that:

$$\lambda_1^{noisy} = \frac{(\theta_1 + 2\theta_2) + \sqrt{{\theta_1}^2 + 4{\theta_2}^2}}{2} + \epsilon_1 \tag{3.30}$$

$$\lambda_2^{noisy} = \frac{(\theta_1 + 2\theta_2) - \sqrt{{\theta_1}^2 + 4{\theta_2}^2}}{2} + \epsilon_2 \tag{3.31}$$

whereby the "noise" terms, ϵ_1 and ϵ_2, both follow a Normal distribution with means 0.0 and standard deviations 1.0 and 0.5 respectively. For this problem, 15 independent "noisy" data from each model is simulated and will be used to perform the analysis. The available data are presented in the form of a scatterplot shown in Figure 3.18 while its numerical values are presented in a table as shown in Table 3.6.

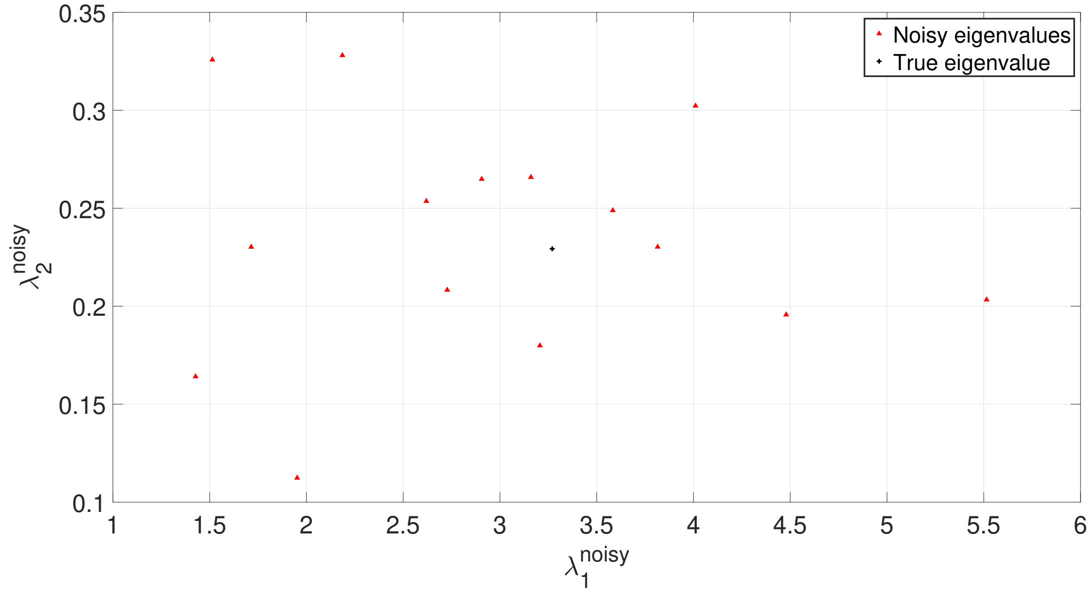


Figure 3.18: Scatterplot of the 15 different measured values of λ_1^{noisy} and λ_2^{noisy}.

Measurement no.	λ_1^{noisy}	λ_2^{noisy}	Measurement no.	λ_1^{noisy}	λ_2^{noisy}
1	1.51	0.33	9	1.95	0.11
2	4.01	0.30	10	4.48	0.20
3	3.16	0.27	11	1.43	0.16
4	3.21	0.18	12	2.91	0.26
5	2.19	0.33	13	3.81	0.23
6	1.71	0.23	14	3.58	0.25
7	2.73	0.21	15	2.62	0.25
8	5.51	0.20	–	–	–

Table 3.6: Numerical values of the "noisy" data illustrated in Figure 3.18.

3.5.1 Bayesian model updating

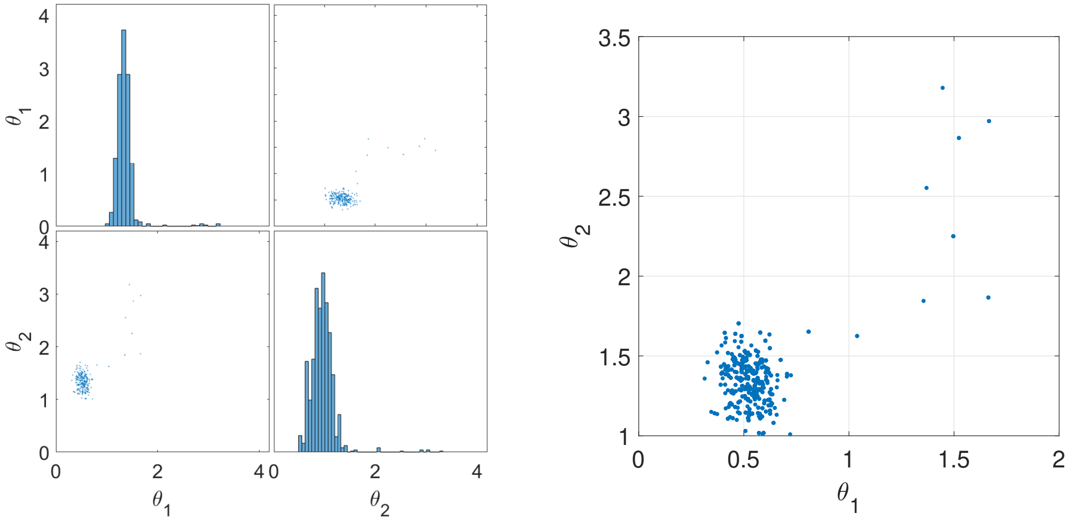
For this problem, the a-priori knowledge of θ_1 and θ_2 is that they both can take values between 0.01 and 4. As such, the prior distribution $P(\theta_1, \theta_2|M)$ (i.e. M is represented by the models for λ_1 and λ_2) can be modelled after a $2D$ Uniform distribution whose lower-bound and upper-bound values are 0.01 and 4 respectively in both dimensions. The likelihood function is modelled using a $2D$ Normal distribution whose covariance matrix has off-diagonal element equal to 0 and diagonal elements corresponding to the standard deviation of each of the "noise" terms, ϵ_1 and ϵ_2. Thus, the $2D$ likelihood function is expressed as follows:

$$P(\boldsymbol{\lambda}|\boldsymbol{\theta}, M) \propto \exp\left[-\frac{1}{2}\sum_{d=1}^{2}\sum_{n=1}^{15}\left(\frac{\lambda_{d,n}^{noisy}-\lambda_{d}^{model}}{\sigma_d}\right)^2\right] \tag{3.32}$$

whereby $\boldsymbol{\lambda}$ is the 15 by 2 vector of the "noisy" observations, $\boldsymbol{\theta} = (\theta_1, \theta_2)$ is the vector of the uncertain model parameters, and σ_i is the standard deviation of ϵ_d, for $d = 1, 2$.

3.5.2 MCMC sampler results

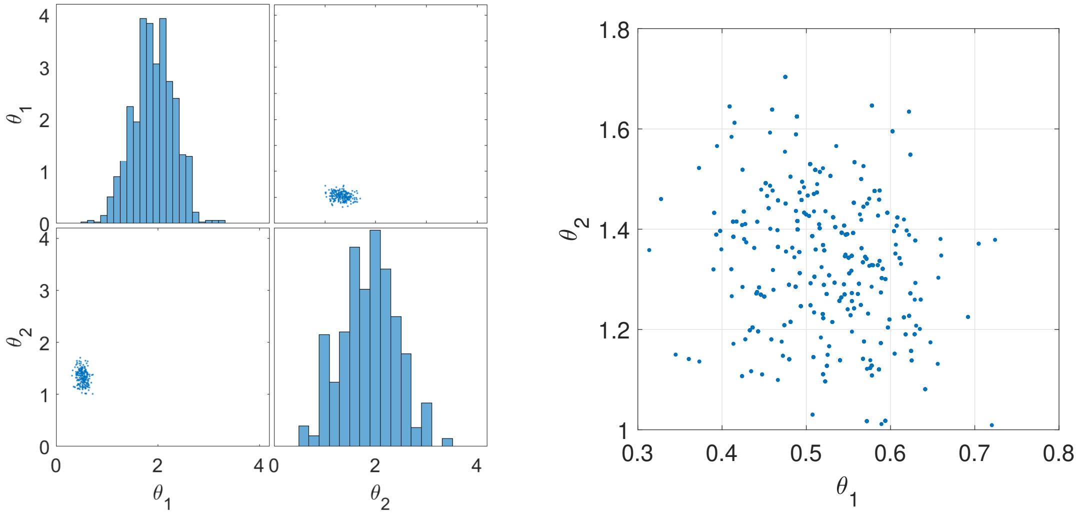
The main sampling procedure is performed with a sample size of $N = 1060$ to ensure sufficient convergence of the sample estimate of θ_1 and θ_2 by the MCMC sampler. The tuning parameter for the sampler is set at $0.04 \cdot I$, where I denotes the Identity matrix. This yields an acceptance level of 0.237. The chain is initiated at $\{\theta_1, \theta_2\} = \{2.84, 2.33\}$ and sampling is first done with 0 burn-in. The resulting scatterplot matrix and $2D$ scatterplot are presented in Figure 3.19. To ensure sufficient burn-in, the burn-in length is set to be 60 and the resulting scatterplot matrix and $2D$ scatterplot as shown in Figure 3.20 where it can be observed that the samples converge about $\{\theta_1,\, \theta_2\} = \{0.51,\, 1.36\}$. The sampler took 0.41 seconds of computation time over 1060 iterations.



(a)

(b)

Figure 3.19: The resulting scatterplot matrix (a) and $2D$ scatterplot (b) obtained using the MCMC sampler with sample size $N = 1060$ and $N_{burn-in} = 0$.



(a) (b)

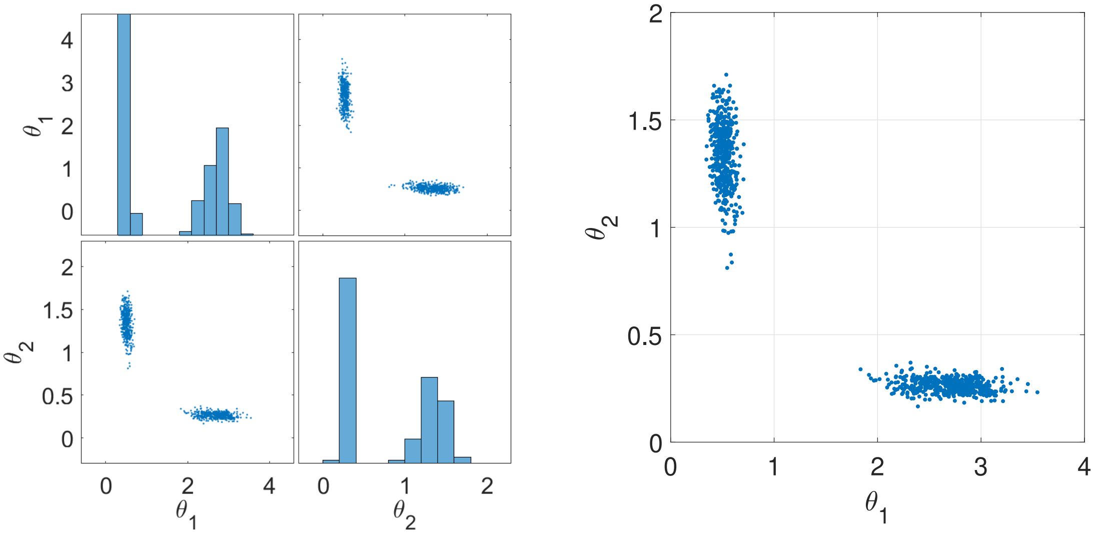
Figure 3.20: The resulting scatterplot matrix (a) and $2D$ scatterplot (b) after discarding the first $N_{burn-in} = 60$ samples.

3.5.3 TMCMC sampler results

For the TMCMC sampler, a sample size of $N = 1000$ samples was obtained from the posterior distribution to ensure sufficient convergence of the sample estimate of θ_1 and θ_2. The sampler took 18.38 seconds of computation time over 5 iterations and the resulting scatterplot matrix and $2D$ scatterplot are presented in Figure 3.21. Based on Figure 3.21, the scatterplot features 2 distinct convergence points centered about $\{\theta_1, \theta_2\} = \{0.51, 1.35\}$ and $\{2.75, 0.27\}$.

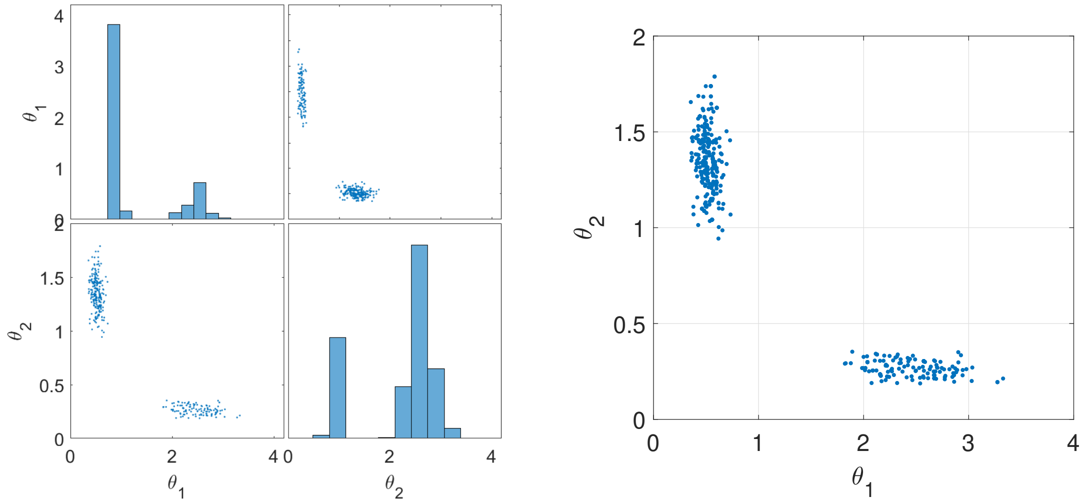
3.5.4 SMC-MH sampler results

For the SMC-MH sampler, a sample size of $N = 1000$ samples was obtained from the posterior distribution to ensure sufficient convergence of the sample estimate of θ_1 and θ_2. The sampler took 6.30 seconds of computation time over 1 iterations and the resulting scatterplot matrix and $2D$ scatterplot are presented in Figure 3.21. Based on Figure 3.22, the scatterplot features 2 distinct convergence points centered about $\{\theta_1, \theta_2\} = \{0.53, 1.38\}$ and $\{2.73, 0.27\}$.



(a) (b)

Figure 3.21: The resulting 2D scatterplot matrix (a) and scatterplot (b) obtained using the TMCMC sampler with sample size $N = 1000$.



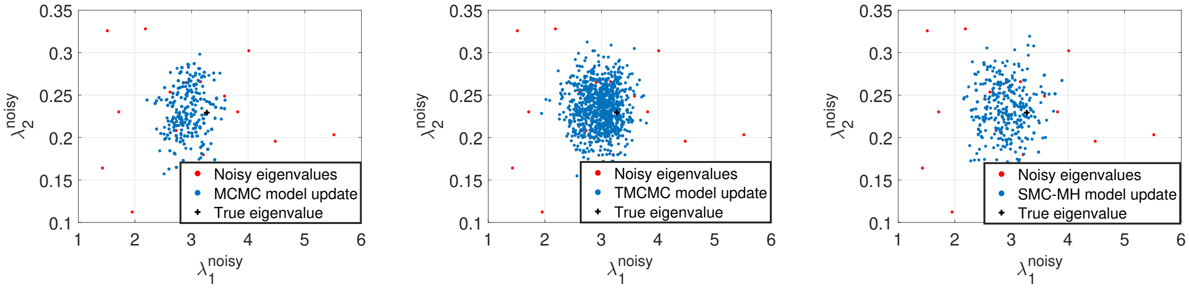
(a) (b)

Figure 3.22: The resulting 2D scatterplot matrix (a) and scatterplot (b) obtained using the SMC-MH sampler with sample size $N = 1000$.

3.5.5 Discussions

From the above results, it can be seen that the MCMC sampler is only able to identify 1 out of the 2 peaks of the bi-modal posterior. This is attributed to the acceptance criteria of the MH algorithm as described in Section 3.1 which results in the samples converging to only one of the modes of a multi-modal posterior. As such, the MCMC sampler would not be a suitable choice of sampler to sample for such posteriors. This shortcoming of the MCMC sampler has also been reflected in [198] when discussing the implementation of the algorithm on a posteriors with multiple peaks. However, should the MCMC sampler be allowed to run infinitely (i.e. generate an infinite number of samples), the resulting distribution of the samples obtained would eventually converge to that of the true posterior and would reflect the peaks of the posterior distribution. The TMCMC and the SMC-MH samplers on the other hand are able to identify all the peaks of the bi-modal posterior. For the TMCMC sampler, this is attributed to the use of the transitional distributions which ensures that the samples are evenly sampled across the sample space from the prior to the posterior as described in Section 3.2. For the SMC-MH sampler, this is attributed to the Importance sampling and resampling procedure which ensures that the samples close to the peaks of the posterior are resampled with higher probability according to Eq. (3.20). This allows for the updated samples to converge towards both peaks.

Based on the estimation results of θ_1 and θ_2 identified by the respective samplers, the updated model using the MCMC, TMCMC, and SMC-MH samplers are presented in Figure 3.23.



(a) MCMC model update.

(b) TMCMC model update

(c) SMC-MH model update

Figure 3.23: Updated scatterplot profiles obtained from: (a) MCMC, (b) TMCMC, and (c) SMC-MH sampling methods.

3.6 Case study 3: DLR-AIRMOD test structure

In order to investigate the existing variability of the natural response frequency in the dynamic behaviour of nominally identical test structures, the Göttingen's German Aerospace Centre (DLR) constructed a replica of the GARTEUR SM-AG19 benchmark known as AIRcraft MODel (AIRMOD) [199]. The DLR-AIRMOD is an aluminum structure consisting of 6 aluminium beams connected using 5 bolted joints. Geometrically, it has a wingspan of 2.0 m, a fuselage length of 1.5 m, and a height of 0.46 m. The structure itself weighs 44.0 kg. At the forward tips of the winglets, 2 additional masses, each weighing 0.167 kg, are installed so as to obtain better excitation of the wing torsions. To represent the free-free boundary conditions, the entire DLR-AIRMOD structure is supported on soft bungee cords. Due to copyright reasons, illustrative photos and schematic diagrams to the DLR-AIRMOD structure could not be provided in the dissertation. Instead, readers may refer to the following references: [2, 200, 3]

The model updating procedure was done using a test data set of frequency response functions. These measurements are obtained from an experiment which involves disassembling and reassembling the structure 130 times to produce 260 different modal data sets from single point excitation at 2 locations for variability. To prevent further variability in the experimental data, the accelerometers remained installed on the structure to ensure their positions remained unchanged during the disassembling and reassembling procedure. In that experiment, 18 input parameters were identified and selected to represent the variability associated with the position of the glue, screws, and cable bundles in the DLR-AIRMOD structure each time after it was reassembled. Details of the 18 input parameters and their respective nominal values are summarised in Table 3.7.

From the experiment campaign, the frequency response functions are obtained. Through the use of experimental modal analysis, 30 different vibration modes and its respective frequencies were obtained of which 14 of them are identified as "active modes". These active modes will be used as measurement outputs for model updating. Their respective details and test statistics are summarised in Table 3.8. In a previous work by Patell *et. al* (2017) [3], a deterministic model updating via the sensitivity method [201, 202] was performed using the information provided in Table 3.8 to update the 18 input uncertain model parameters listed in Table 3.7. The resulting statistics of the 18 updated parameters are summarised in Table 3.9.

In this section, the Bayesian model updating approach is be adopted to update the 18

$\boldsymbol{\theta}$	Type	Location	Description	Init. val.	Unit
θ_1	Stiffness	Front Bungee Cord	Support Stiffness	1.80×10^3	N/m^2
θ_2	Stiffness	Rear Bungee Cord	Support Stiffness	7.50×10^3	N/m^2
θ_3	Stiffness	VTP/HTP Joint	Sensor Cable - y dir^n	1.30×10^2	N/m
θ_4	Stiffness	Wing/Fuselage Joint	Top Sensor Cable - y dir^n	7.00×10^1	N/m
θ_5	Stiffness	Wing/Fuselage Joint	Bottom Sensor Cable - y dir^n	7.00×10^1	N/m
θ_6	Stiffness	VTP/HTP Joint	Joint Stiffness - x, y dir^{ns}	1.00×10^7	N/m
θ_7	Stiffness	VTP/HTP Joint	Joint Stiffness - z dir^n	1.00×10^9	N/m
θ_8	Mass	VTP/HTP Joint	Sensor Cables	2.00×10^{-1}	kg
θ_9	Mass	Wingtip Right Wing	Screws and Glue	1.86×10^{-1}	kg
θ_{10}	Mass	Wingtip Left Wing	Screws and Glue	1.86×10^{-1}	kg
θ_{11}	Mass	Wingtip Left/Right	Sensor Cables on Wings	1.50×10^{-2}	kg
θ_{12}	Mass	Outer Wing Left/Right	Sensor Cables on Wings	1.50×10^{-2}	kg
θ_{13}	Mass	Inner Wing Left/Right	Sensor Cables on Wings	1.50×10^{-2}	kg
θ_{14}	Stiffness	Wing/Fuselage Joint	Joint Stiffness - x dir^n	2.00×10^7	N/m
θ_{15}	Stiffness	Wing/Fuselage Joint	Joint Stiffness - y dir^n	2.00×10^7	N/m
θ_{16}	Stiffness	Wing/Fuselage Joint	Joint Stiffness - z dir^n	7.00×10^6	N/m
θ_{17}	Stiffness	VTP/Fuselage Joint	Joint Stiffness - x dir^n	5.00×10^7	N/m
θ_{18}	Stiffness	VTP/Fuselage Joint	Joint Stiffness - y dir^n	1.00×10^7	N/m

Table 3.7: List of the 18 input parameters and their respective details. Data obtained from [2].

Output	Mode name	f_{mean} $[Hz]$	f_{std} $[Hz]$	COV [%]	Sample size
f_1	RBM Yaw	0.23	0.006	2.41	41
f_2	RBM Roll	0.65	0.019	2.89	81
f_3	RBM Pitch	0.83	0.017	1.99	83
f_4	RBM Heave	2.17	0.024	1.11	86
f_5	2nWingBending	5.50	0.004	0.07	86
f_6	3nWingBending	14.91	0.017	0.05	86
f_7	WingTorsionAnti	31.96	0.020	0.06	86
f_8	WingTorsionSym	32.33	0.017	0.05	86
f_9	4nWingBending	43.89	0.015	0.03	86
f_{10}	1nWingForeAft	46.71	0.149	0.32	86
f_{11}	2nWingForeAft	51.88	0.012	0.02	86
f_{12}	VtpTorsion	65.93	0.274	0.42	86
f_{13}	2nHtpBending	205.59	1.023	0.50	86
f_{14}	HtpForeAft	219.07	1.663	0.76	86

Table 3.8: Test statistics of the 14 frequency outputs to be used to perform model updating. Data obtained from [2].

$\boldsymbol{\theta}$	Mean	Std	COV [%]	Unit
θ_1	1.82×10^3	1.08×10^2	5.94	N/m^2
θ_2	7.90×10^3	2.40×10^2	3.04	N/m^2
θ_3	1.87×10^2	1.09×10^1	5.85	N/m
θ_4	4.47×10^1	2.03×10^0	4.55	N/m
θ_5	4.24×10^1	2.24×10^0	5.29	N/m
θ_6	2.53×10^6	3.50×10^5	13.83	N/m
θ_7	7.80×10^8	2.56×10^8	32.82	N/m
θ_8	1.86×10^{-1}	7.60×10^{-3}	4.08	kg
θ_9	2.09×10^{-1}	4.65×10^{-3}	2.22	kg
θ_{10}	1.90×10^{-1}	4.28×10^{-3}	2.26	kg
θ_{11}	3.00×10^{-2}	1.26×10^{-3}	4.20	kg
θ_{12}	9.83×10^{-3}	1.22×10^{-3}	12.37	kg
θ_{13}	1.47×10^{-2}	1.65×10^{-4}	1.12	kg
θ_{14}	4.07×10^7	1.32×10^6	3.24	N/m
θ_{15}	9.48×10^6	1.06×10^6	11.18	N/m
θ_{16}	2.93×10^6	1.89×10^5	6.44	N/m
θ_{17}	8.75×10^6	2.80×10^6	32.00	N/m
θ_{18}	5.97×10^6	6.90×10^5	11.56	N/m

Table 3.9: Updated statistics of the 18 input parameters obtained using the Sensitivity model updating method. Results taken from [3].

input parameters which will be done using the MCMC, TMCMC and SMC-MH sampling techniques. The purpose of this is to compare the sampling and model updating performances of each of the samplers as well as to assess and highlight the robustness of each algorithm in sampling from a relatively complex, higher-dimensional posterior. The model to be used for the Bayesian updating procedure is a surrogate model in the form of an Artificial Neural Network (ANN). It serves to provide a relatively computationally inexpensive approach compared to the finite element modelling and is assumed to be an accurate representation of the underlying model. An ANN model used to predict the 14 response frequencies of the test structure was previously developed by Patelli *et. al* (2017) [80] and will be implemented in the analysis here. The ANN model consists of 14 individual ANNs, each trained to predict a specific frequency output of the corresponding active mode [80]. Each of these 14 individual ANN comprises of 1 input layer consisting of 18 nodes, 2 hidden-layers consisting of 16 nodes in the first and 6 nodes in the second respectively, and 1 output layer consisting of 1 node. For each of these ANNs, a sigmoid activation function is used in the form of the hyperbolic tangent function [80]. In this example, the simulation and computation was implemented using the OpenCossan software [203, 204].

3.6.1 Bayesian model updating

Contrary to the standard procedure of using prior identified from previous experience or methods (see e.g. [3]), the Uniform prior is used here. Therefore, for each of the individual input parameter, θ_d (for $d = 1, ..., 18$), $P(\theta_d|M)$, is modelled as a Uniform distribution whose lower bound and upper bound is 5 % and 200 % of the input's nominal values, respectively. The input parameters are assumed to be independent of one another and thus, the overall prior distribution, $P(\boldsymbol{\theta}|\boldsymbol{M})$, can be expressed as follows:

$$P(\boldsymbol{\theta}|M) = \prod_{d=1}^{18} P(\theta_d|M) \tag{3.33}$$

The likelihood function is a 14-dimensional multivariate Normal distribution. Assuming independence between the experimental outputs, it is mathematically expressed as follows [3]:

$$P(\boldsymbol{f}|\boldsymbol{\theta}, M) \propto \exp\left[-\sum_{d=1}^{14} \frac{(f_d - M_d)^2}{2 \cdot \sigma_d^2}\right] \tag{3.34}$$

whereby f_d is the experimental measurement of the d^{th} frequency mode, M_d is the d^{th} ANN model used to predict the frequency output of the d^{th} active mode, and σ_d is the standard deviation of the residual between the experimental result f_d and the model $\hat{M}_d$. Here, σ_d is not a fixed constant parameter unlike in the previous examples. Instead, this hyper-parameter will be inferred directly from the residual between experimental data values and the ANN model.

In this problem, the sampling performances of the MCMC, TMCMC, and SMC-MH samplers are compared on the basis of: 1) their effectiveness in sampling from the relatively high-dimensional posterior; and 2) their respective Monte Carlo standard error σ_{SE} on the estimation of $\boldsymbol{\theta}$ which is computed from Eq. (3.4) using batch size $n_b = 100$.

3.6.2 MCMC sampler results

A nominal covariance matrix $\boldsymbol{\Sigma}$ is first constructed in the form of a diagonal matrix whose diagonal elements correspond to the respective variance of the posterior for each of the 18 input parameters. This is done using the information from statistics of the updated input parameters obtained using the sensitivity method which is presented in Table 3.9. To ensure that the acceptance rate of the sampler is within the acceptable range of 0.15 to 0.50, the tuning parameter is set at $10^{-3} \cdot \mathbf{\Sigma}$. A sample size of $N = 1500$ is obtained from the posterior, with a burn-in length of $N_{burn-in} = 500$, and the simulation was performed on 1 core with a CPU memory of 10.5 Gigabytes. The computation involved a total of 1500 iterations over 83.81 seconds with an acceptance rate of 0.258. The resulting statistics of the updated input parameters are summarised in Table 3.10, while the posterior distribution of the normalised data for each input parameter is presented as a scatterplot matrix illustrated in Figure 3.24.

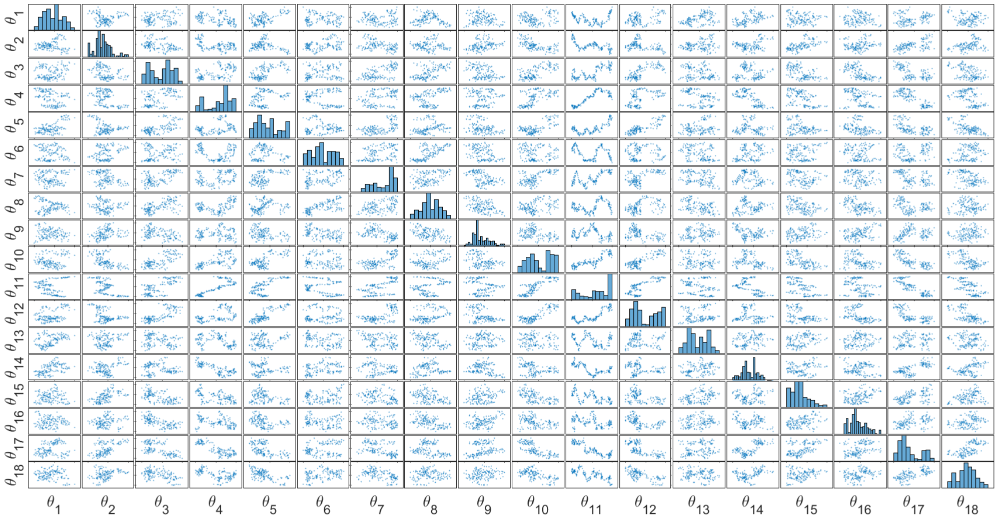


Figure 3.24: Scatterplot matrix of the 18 inputs obtained using the MCMC sampler with $N = 1500$ samples and $N_{burn-in} = 500$. The data presented here are normalised to take values between 0 and 1.

3.6.3 TMCMC sampler results

Using the TMCMC sampler, a sample size of $N = 1000$ samples was obtained from the posterior. The simulation was performed using local parallelisation across 34 cores, each with a CPU memory of 10.5 Gigabytes. The computation involved a total of 22 iterations over 5 hours 13 minutes. The resulting statistics of the updated input parameters are summarised in Table 3.10, while the posterior of the normalised data for each input parameter are presented in a scatterplot matrix as illustrated in Figure 3.25.

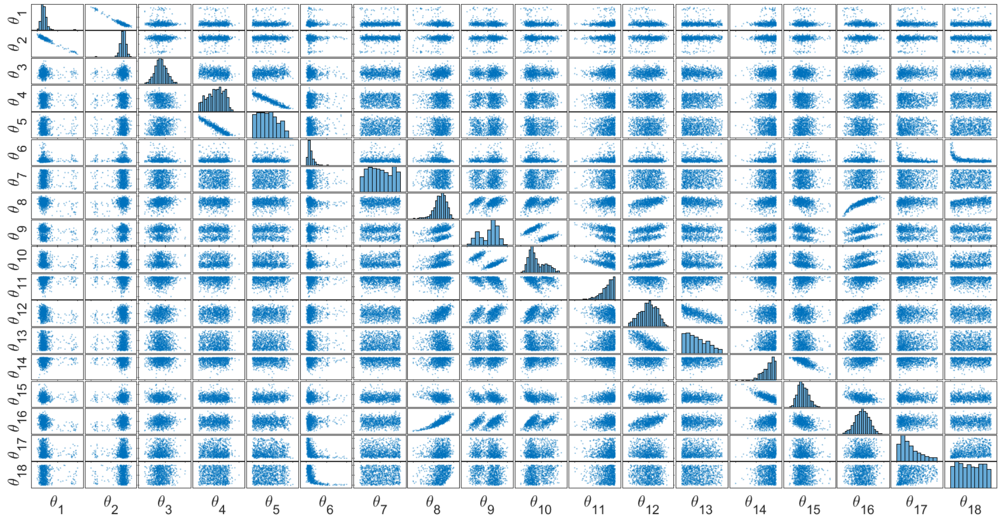


Figure 3.25: Scatterplot matrix of the 18 inputs obtained using the TMCMC sampler with $N = 1000$ samples. The data presented here are normalised to take values between 0 and 1.

3.6.4 SMC-MH sampler results

Using the SMC-MH sampler, a sample size of $N = 1000$ samples was obtained from the posterior. The simulation was performed using local parallelisation across 15 cores, each with a CPU memory of 10.5 Gigabytes. The computation involved a total of 1 iteration over 17 hours 53 minutes which is significantly longer than that of the TMCMC and the MCMC sampling algorithms. This is attributed to the sub-optimal implementation of the SMC-MH sampler in the dissertation. The resulting statistics of the updated input parameters are summarised in Table 3.10, while the posterior of the normalised data for each input parameter are presented in a scatterplot matrix as illustrated in Figure 3.26.

3.6.5 Discussions

Table 3.10 summarises the numerical results of the estimates obtained by the respective samplers. As a form of evaluating the model updating performance between the 3 samplers, the scatterplot profiles illustrating the distribution of the experimental samples will be compared against those from the the updated ANN surrogate model as seen in Figures 3.27, 3.28, and 3.29.

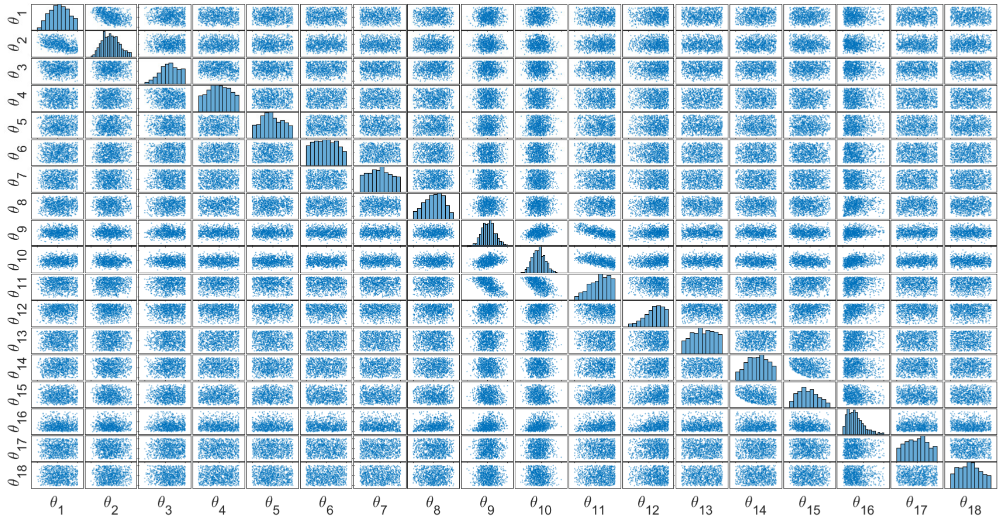


Figure 3.26: Scatterplot matrix of the 18 inputs obtained using the SMC-MH sampler with $N = 1000$ samples. The data presented here are normalised to take values between 0 and 1.

To quantify the closeness and similarity level between the marginal distributions of the experimental frequency samples and the sample output obtained from the updated model, the 2-sample univariate Kolmogorov–Smirnov (KS) test [205, 206, 207] is used as the metric. The 2-sample KS test tests the null hypothesis that 2 given samples come from the same continuous distribution against the alternative hypothesis that they do not. The 2-sided test is performed at 5 % significance level for each of the 14 frequency outputs. Table 3.11 presents the resulting p-values as well as the logical value of the test indicator where 0 indicates that there is insufficient evidence to reject the null hypothesis, and 1 indicates that there is sufficient evidence to reject the null hypothesis at 5 % level of significance.

For the case of the MCMC sampler, Table 3.11 shows that the 2-sample KS test indicates that there is no similarity between the marginal distributions of the experimental frequency samples and the sample output from the updated model for all 14 active frequencies tested. This is supported from Figure 3.27 where it can be seen that frequency scatterplots profile from the updated model (in blue) obtained using MCMC sampling technique do not show any similarity to that of the experimental frequencies (in red). Such observations indicate that the MH sampling algorithm is unable to perform Bayesian model updating effectively which demonstrates its limitation in sampling from a high-dimensional posterior distribu-

$\boldsymbol{\theta}$	MCMC		TMCMC		SMC-MH	
	Mean	σ_{SE}	Mean	σ_{SE}	Mean	σ_{SE}
θ_1	1.13×10^3	3.46	1.89×10^3	8.00	2.05×10^3	20.17
θ_2	8.98×10^3	6.76	7.64×10^3	21.26	7.78×10^3	70.35
θ_3	1.27×10^2	3.33	1.85×10^2	0.27	1.63×10^2	2.32
θ_4	5.70×10^1	0.14	5.25×10^1	0.79	7.25×10^1	1.04
θ_5	3.52×10^1	0.14	3.59×10^1	0.65	6.98×10^1	0.99
θ_6	1.20×10^7	1.63×10^4	1.47×10^6	2.18×10^4	9.85×10^6	2.04×10^5
θ_7	1.32×10^9	7.80×10^6	1.03×10^9	1.32×10^7	1.00×10^9	1.85×10^7
θ_8	2.49×10^{-1}	2.08×10^{-4}	1.92×10^{-1}	8.79×10^{-4}	2.14×10^{-1}	2.10×10^{-3}
θ_9	4.07×10^{-2}	8.79×10^{-5}	1.99×10^{-1}	5.94×10^{-4}	2.39×10^{-1}	1.10×10^{-3}
θ_{10}	2.48×10^{-1}	1.50×10^{-4}	1.95×10^{-1}	4.69×10^{-4}	2.40×10^{-1}	1.00×10^{-3}
θ_{11}	1.13×10^{-2}	2.73×10^{-4}	2.82×10^{-2}	3.96×10^{-5}	1.90×10^{-2}	2.68×10^{-4}
θ_{12}	1.21×10^{-2}	9.76×10^{-5}	1.03×10^{-2}	1.13×10^{-4}	1.95×10^{-2}	1.71×10^{-4}
θ_{13}	2.81×10^{-2}	4.26×10^{-6}	1.17×10^{-2}	1.97×10^{-4}	1.62×10^{-2}	3.03×10^{-4}
θ_{14}	1.89×10^7	3.91×10^4	3.61×10^7	8.49×10^4	2.18×10^7	4.57×10^5
θ_{15}	2.72×10^7	2.57×10^4	1.04×10^7	5.13×10^4	1.89×10^7	3.36×10^5
θ_{16}	6.99×10^6	4.08×10^3	2.36×10^6	1.68×10^4	4.75×10^6	1.16×10^5
θ_{17}	1.00×10^7	1.06×10^5	3.49×10^7	7.96×10^5	5.12×10^7	1.11×10^6
θ_{18}	4.24×10^7	1.68×10^8	5.01×10^7	1.28×10^6	4.92×10^7	7.71×10^5

Table 3.10: Updated statistics of the 18 input parameters obtained using the MCMC, TMCMC, and SMC-MH samplers.

tion, especially when the posterior is only concentrated within a small area of the entire sample space. This comes despite the Monte Carlo standard error σ_{SE} of the estimation of the 18 input parameters are less than 10 % of the corresponding mean estimate values as seen in Table 3.10. In addition, the efficiency of the MCMC sampler depends on the choice of the user-defined tuning parameter, making such sampler an unfavourable choice for such problem. For this study, the tuning parameter was defined based on the results obtained using Sensitivity model updating [3] (see Table 3.9) whereby the information on the standard deviation of each of the 18 updated input parameters was used to construct the nominal covariance matrix to begin with. Should such prior information be unavailable, the tuning parameter may have to be determined via "Trial-and-Error" which will be extremely inefficient and impractical for such high-dimensional problem. Furthermore, it needs to be highlighted that the posterior samples in Figure 3.24 shows a high auto-correlation for the case of θ_{11}. This is attributed to the choice of the tuning parameter used in this study, to which the justification of its use for this problem is explained earlier, may not be optimal.

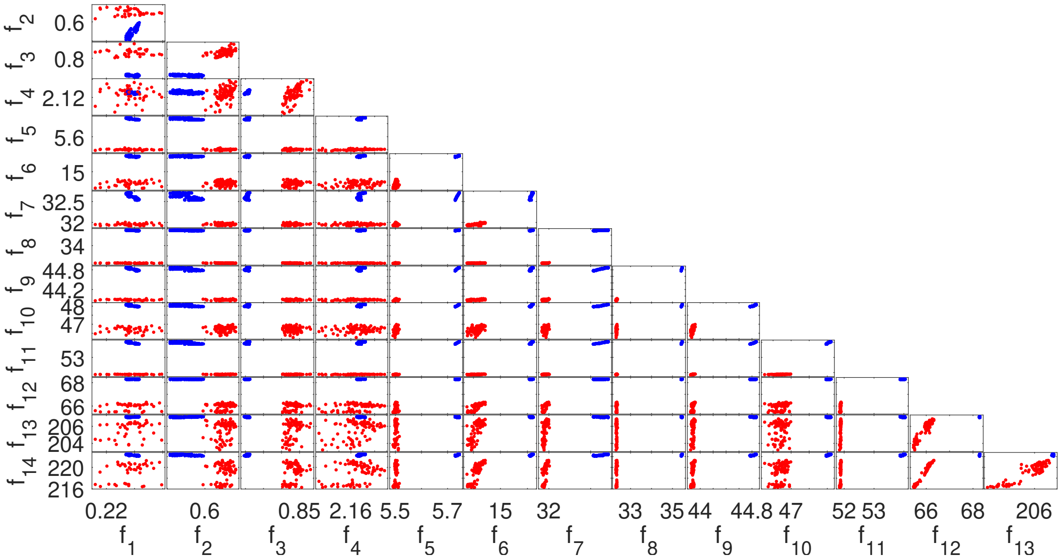


Figure 3.27: The scatterplot matrix illustrating the updated model output profile obtained using the MCMC sampling technique. The blue scatterplots represent the frequency output from the updated model while the red scatterplots represent the experimental frequency measurements.

Output	Mode name	MCMC		TMCMC		SMC-MH	
		p-value	Indicator	p-value	Indicator	p-value	Indicator
f_1	RBM Yaw	6.73×10^{-5}	1	2.67×10^{-1}	0	1.24×10^{-23}	1
f_2	RBM Roll	2.19×10^{-67}	1	1.10×10^{-1}	0	1.17×10^{-18}	1
f_3	RBM Pitch	5.03×10^{-69}	1	1.52×10^{-1}	0	5.99×10^{-14}	1
f_4	RBM Heave	8.13×10^{-17}	1	7.19×10^{-4}	1	6.72×10^{-15}	1
f_5	2nWingBending	2.86×10^{-71}	1	1.97×10^{-13}	1	1.83×10^{-54}	1
f_6	3nWingBending	2.86×10^{-71}	1	1.75×10^{-1}	0	1.18×10^{-19}	1
f_7	WingTorsionAnti	2.86×10^{-71}	1	6.84×10^{-2}	0	3.92×10^{-32}	1
f_8	WingTorsionSym	2.86×10^{-71}	1	6.50×10^{-3}	1	5.07×10^{-18}	1
f_9	4nWingBending	2.86×10^{-71}	1	5.13×10^{-1}	0	1.43×10^{-33}	1
f_{10}	1nWingForeAft	2.86×10^{-71}	1	5.24×10^{-8}	1	2.61×10^{-15}	1
f_{11}	2nWingForeAft	2.86×10^{-71}	1	3.06×10^{-58}	1	1.92×10^{-26}	1
f_{12}	VtpTorsion	2.86×10^{-71}	1	4.77×10^{-4}	1	1.69×10^{-64}	1
f_{13}	2nHtpBending	2.86×10^{-71}	1	2.16×10^{-4}	1	5.43×10^{-14}	1
f_{14}	HtpForeAft	2.86×10^{-71}	1	1.53×10^{-5}	1	5.80×10^{-59}	1

Table 3.11: P-values and test indicator from the 2-sample KS test performed on the frequency samples from the updated model by the MCMC, TMCMC, and SMC-MH samplers along with the frequency samples obtained from experiment.

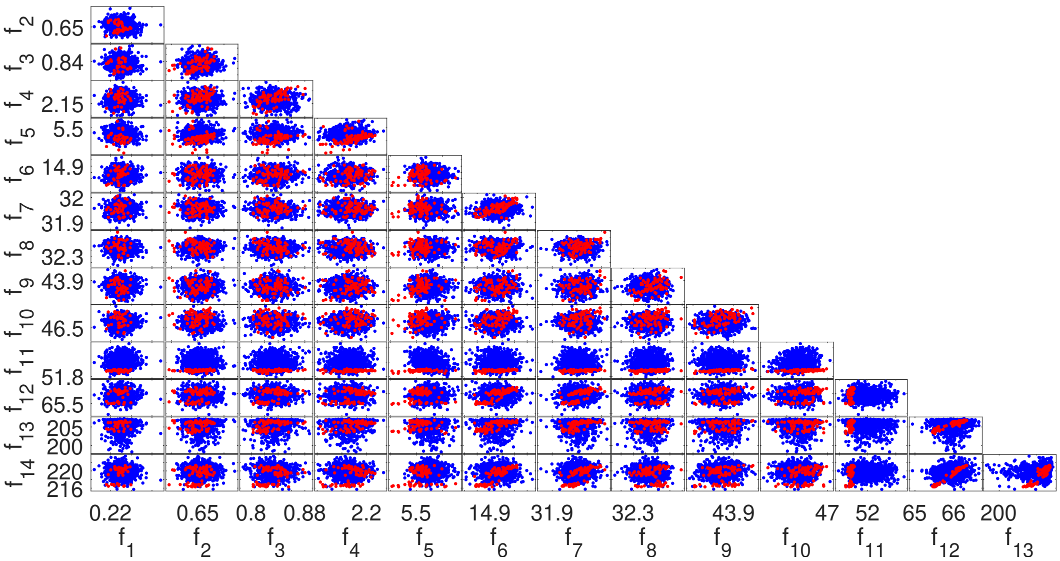


Figure 3.28: The scatterplot matrix illustrating the updated model output profile obtained using the TMCMC sampling technique. The blue scatterplots represent the frequency output from the updated model while the red scatterplots represent the experimental frequency measurements.

For the case of the TMCMC sampler, the KS test indicates that there was insufficient evidence at 5 % significance level to reject the null hypothesis for frequency outputs f_1, f_2, f_3, f_6, f_7, and f_9. This implies that the test failed to identify significant degree of differences between the marginal distributions of the experimental frequency samples and the sample output from the updated model for the aforementioned 6 active frequencies. In addition, it can be observed from Figure 3.28 that the frequency scatterplot profile for the updated model mostly coincides with the frequency scatterplot profile for the experimental frequencies. Coupled with the fact that the algorithm does not require the user to perform any manual tuning of the tuning parameter(s) (i.e. the covariance matrix $\mathbf{\Sigma}^j$), this makes the TMCMC sampler a suitable choice of sampler for such a problem.

For the case of the SMC-MH sampler, Table 3.11 shows that the 2-sample KS test indicates that there is no similarity between the marginal distributions of the frequency samples and the sample output from the updated model for all 14 active frequencies tested. This is observed from Figure 3.29 where it can be seen that while the frequency scatterplots from the updated model generally encompass the frequency scatterplots from the experimental data, there is no similarity in the scatterplot profiles between the 2 entities. In addition,

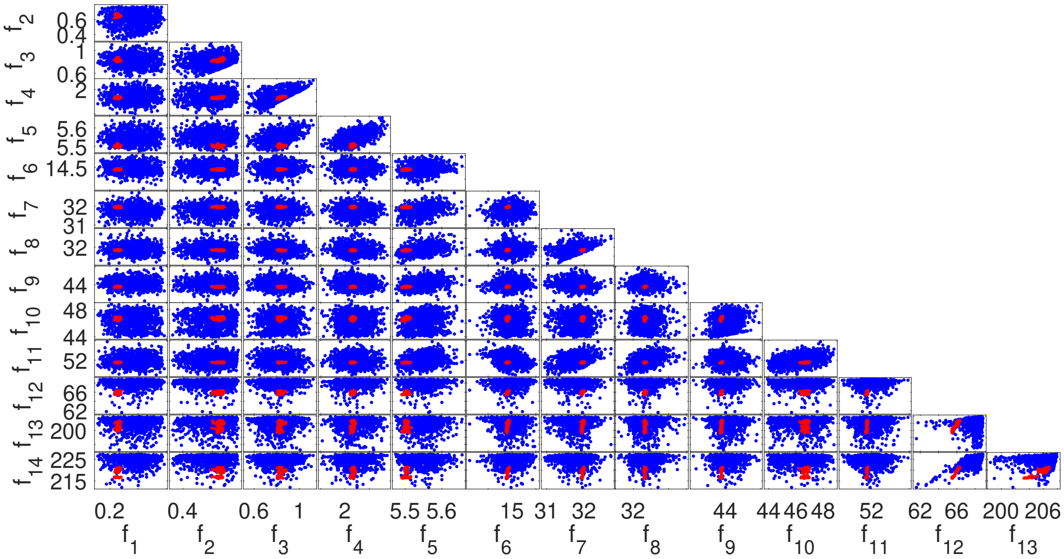


Figure 3.29: The scatterplot matrix illustrating the updated model output profile obtained using the SMC-MH sampling technique. The blue scatterplots represent the frequency output from the updated model while the red scatterplots represent the experimental frequency measurements.

it can also be observed from the figure that the scatterplot profile from the updated model shows a larger spread compared to that of the experimental data. Such is attributed to the estimates of the 18 updated input parameters having relatively higher σ_{SE} (and therefore a higher value of sample standard deviation σ_{θ} according to Eq. (3.4)) values in general as shown in Table 3.10 and that some of the marginal posterior distributions illustrate a small degree of difference from the Uniform prior such as in the case for θ_4, θ_6, θ_7 and θ_{13} in Figure 3.26. This is due to the ineffectiveness in the Importance sampling technique when applied in high-dimensional cases [51, 208, 209], to which an extensive study has been conducted in [210].

It has to be noted that the intention of this experiment was neither to obtain the optimum updating results as such work was previously done and presented in the literature by Govers *et. al* (2015) [2] nor to develop better surrogate model(s) for the Bayesian model updating procedure. Rather, the purpose of this section is to evaluate the model updating performance between the MCMC, TMCMC, and SMC-MH samplers for a relatively high-dimensional system.

3.7 Further discussions

The 3 sampling techniques reviewed in the chapter have been applied to 3 case studies with different inherent challenges. In summary, the MH algorithm is relatively the easiest to implement among the 3 sampling algorithms discussed and it is useful in sampling from target distributions which are known up to a normalizing constant. However, its efficiency is limited by the choice of the proposal distribution or tuning parameter. As discussed in Section 3.1, should the width of the proposal distribution be too large, it may produce many proposed samples which lie outside the domain of the target distribution thus increasing the rejection rate of the samples. Should the width of the proposal distribution be too small, the rejection-rate of the samples become low but this comes at the expense of the need of many iterations before the Markov chain converges to the stationary distribution. In addition, the MH algorithm is shown to be ineffective in sampling from multi-modal posterior [51] due to its acceptance criteria of the proposed samples which results in the samples converging to one of the peaks as seen from the case example presented in Section 3.5. Such problem can be overcomed by allowing the MH sampler to obtain an infinite number of samples which ensures that the sample distribution would converge to the true posterior and identify all the peaks present. Furthermore, the algorithm is also shown to be ineffective in sampling from a high-dimensional posterior whereby each dimension is independent from one another and that the distribution itself is concentrated within a small subspace [51, 3] of the entire sample space as shown in the case example presented in Section 3.6. These short-comings, however, are addressed with recent developments of the algorithm such as: 1) the Adaptive Metropolis-Hastings (AMH) algorithm [168]; 2) the Adaptive Metropolis-within-Gibbs (AMWG) algorithm [211, 212]; 3) the multi-level MCMC approach proposed by Lam *et. al* (2015) [213] which seeks to improve the algorithm's exploration of the sample space by dividing the sampling process into multiple levels; and 4) the TMCMC algorithm [51].

The TMCMC sampler algorithm is a particular variant of the SMC sampler and is based on the MCMC technique. Its key strength lies in the ability of the algorithm to effectively sample from multi-modal posteriors as seen in Section 3.5 as well as high-dimensional posterior as seen in Section 3.6 which makes the TMCMC a robust sampler [51]. In addition, the issue of burn-in is less of a concern given that the initial set of samples obtained in the initialization stage of the algorithm comes directly from the prior which prevents the occurrences of obtaining samples from outside the posterior. One disadvantage however is that due to the relative complexity of the algorithm, the computation time evolved in executing the entire sampling process becomes significantly longer as observed in all the case examples whereby the time elapsed by the TMCMC sampler is consistently the highest among the 3 samplers. This is attributed to the higher number of model evaluations that is done by the algorithm as a result of the need to generate samples from not just the posterior alone, but also from the transitional distributions.

The SMC-MH sampler algorithm does not require the need to consider a burn-in period due to the fact that the algorithm is fundamentally based on the Importance sampling technique, and that the initialization procedure of the algorithm which obtains an initial set of samples directly from the prior distribution ensures that the final samples obtained are within the posterior. In addition, the SMC-MH sampler is also able to sample from a multi-modal posterior shown in Section 3.5 where it managed to identify the 2 peaks of a bi-modal posterior distribution. However, a significant drawback of the SMC-MH sampler implemented in this study is its reduced efficiency and effectiveness in sampling from a high-dimensional posterior [208, 209] as shown in Section 3.6. This can be attributed to the following: 1) the inefficiency and inapplicability of the Importance sampling procedure to samples in high dimensions to which an extensive study can be found in [210]; and 2) the implementation of a sub-optimal Backward Markov kernel $q_L(\boldsymbol{\theta}_i|\boldsymbol{\theta}_i^*)$ as highlighted in Section 3.3.1. To overcome these issues, one can turn to the following: 1) the advanced SMC sampling strategies such as the through the use of an adaptive MCMC mutation kernel proposed in [214], or the nested SMC sampling approach [215, 216]; and 2) the implementation of an approximately optimal Backward Markov kernel introduced in [217].

It is also observed from the case examples that different sampling techniques yield different statistics on the estimates of the inferred parameter(s) (i.e. the mean and standard error). This is due to each sampling method having its own assumption(s) in its respective algorithms. For instance, the MH sampler assumes that by allowing a single Markov chain to continue running for long periods of time, the chain would eventually converge to the stationary distribution corresponding to the posterior distribution [128]. The TMCMC sampler assumes that the samples would eventually converge to the final posterior distribution by sampling from a series of intermediate transitional distributions [51]. This allows of the TMCMC sampler to be able to sample from higher-dimensional posteriors with relatively complicated shapes such as having multiple peaks as seen in Sections 3.5 and 3.6. Finally, the SMC-MH sampler assumes the statistics of the posterior can be approximated through the combination Importance sampling and Resampling procedure [177]. It not only ensures that the samples with insignificant weights are discarded, but also ensures that the samples converge towards regions of the sample space with higher probabilities as defined by the posterior. Such approach, however, may fall short when dealing with high-dimensional posteriors for which reasons are discussed in earlier in Section 3.6 [210, 53]. Thus, this indicates that each of the sampling technique should be chosen depending on the validity of its assumption relative to the problem that needs to be address as well as the computational power that is available.

3.8 Chapter conclusion

Bayesian inference is a popular approach for model updating in engineering applications. Bayesian model updating relies heavily on computational techniques to sample from a posterior distribution. In the chapter, the concept behind the following 3 sampling techniques are presented: Markov Chain Monte Carlo, Transition Markov Chain Monte Carlo, and Sequential Monte Carlo sampler. The presented algorithms have been implemented to solve 3 different engineering problems of increasing difficulty to assess their respective computational performances and robustness. From the case studies presented, it can be seen that different sampling techniques yield different results of the posterior mean and variance due to the different assumptions made in the sampling algorithm as explained in Section 3.7. In addition, it can also be observed that the TMCMC algorithm is the most robust amongst the three samplers given that it is consistently able to sample from posteriors ranging from a simple one-dimensional case, to a more complex 18-dimensional case. The trade-off however comes with its relatively long computation time due to its increased model evaluations as a result of the need to generate samples for every transitional distribution.

While the case studies presented in the chapter are set in the context whereby measurement data set are considered as a single piece of information made available to make inferences on time-invariant uncertain model parameter(s), such conditions are specific and may not necessarily be true at all times. In general, the recorded measurements or data can come at different time-steps, especially when they are obtained from a system that is evolving with time (see [1, 218, 82, 181, 219, 220]). These time-evolving data are related to external factors evolving with time such as the measurement noise ϵ or environmental loading conditions.

Moreover, the parameters to be inferred might be time-varying for example because of degradation effects of the structural materials. In this case, these parameters would conform to a non-stationary posterior distributions which has not been addressed in the chapter. Some approaches implemented to deal with such problems in engineering can be found in references [221, 222, 223]. In addition to this, the recorded data themselves may not necessarily be independently identically distributed, as it was assumed in this study. Furthermore, to broaden the generality of problems that could be encountered, the model relating the measured variables $\boldsymbol{D}$ and the uncertain model parameter(s) $\boldsymbol{\theta}$ may not even be known precisely (i.e. model uncertainty; see [224, 87, 88, 225]). These are currently active research areas.

For the benefit of the readers, the presented algorithms (along with its implementations) and the examples discussed in the chapter are freely available as part of the OpenCossan software [203, 204] on GitHub: `https://github.com/cossan-working-group/BayesianModelUpdating`

Chapter 4

Transitional Ensemble Markov Chain Monte Carlo: A Robust and Efficient sampler for Off-line Bayesian Model Updating

Abstract

The chapter presents the Transitional Ensemble Markov Chain Monte Carlo sampler. The proposed sampler utilises the Affine-invariant Ensemble sampler in place of the classical Metropolis-Hastings sampler as the Markov Chain Monte Carlo move kernel. This allows for the sampling of badly-scaled and highly-anisotropic distributions without requiring extra computational costs. In addition, the weighted random sampling approach is adopted for resampling which allows for more Markov chains to initiate from samples with higher weights, leading to increased number of chains in sample regions of high probability. Furthermore, an adaptive tuning algorithm is also proposed within the new sampler which allows for automatic tuning of the step-size of the Affine-invariant Ensemble sampler. Hence, such proposals not only ensure that the new sampler is "tune-free" for the users, but also improves its robustness by ensuring that the acceptance rate of samples is well-controlled within acceptable bounds. The proposed approach is found to be significantly faster compared to standard Transitional Markov Chain Monte Carlo methods, especially on badly scaled and highly skewed distributions which can be encountered when dealing with complex engineering

problems. The proposed sampler will be implemented on 2 benchmark numerical examples of varying complexities to demonstrate its strengths and advantages.

4.1 Background

In Chapter 3, an introduction and review of the TMCMC sampling algorithm is presented [46]. Recently, it has been applied in numerous engineering problems to which a list of examples is provided in Section 3.2.3. The vast application of the TMCMC sampler over the years highlights its popularity in engineering applications [13, 226]. This is attributed to its capability in: 1) estimating large number of parameters at one time (e.g. 24 model parameters [227]); 2) sample from complex-shaped distributions [51, 172]; and 3) its ability to quantify the suitability of a model in describing the observed data under uncertainty [228, 79, 173, 95, 229].

However, despite such strengths, the TMCMC algorithm still presents numerous short-comings including: 1) having large number of auxiliary parameters to tune due to the choice of proposal distribution [169]; 2) computationally expensive due to the need to re-compute the auxiliary parameters at every iteration [46, 169]; and 3) does not provide a mechanism to control the acceptance rates of the samples within the acceptable bounds of $[0.15, 0.50]$ [150]. To address those short-comings, an improved TMCMC approach called the Transitional Ensemble Markov Chain Monte Carlo (TEMCMC) sampler, is proposed.

The approach employs the Affine-invariant Ensemble sampler (AIES) proposed by Goodman and Weare (2009) [5] in place of the Metropolis-Hastings (MH) sampler for the MCMC step. By using a sampler that satisfies the affine-invariance property, the performance of the method becomes independent from the complexity of the distribution [5]. The motivations behind the proposed implementation are: 1) the AIES can sample efficiently from highly-skewed distribution functions [5]; and 2) having a reduced the number of auxiliary parameters to tune and the computational cost [5].

Another implemented change is in the sample updating strategy. Contrary to the approach that is implemented in the TMCMC sampler, as described in Chapter 3, samples with higher statistical weights associated with it (i.e. see Eq. (3.7)) are more likely to be assigned as the starting sample of the Markov chain, leading to more Markov chains initiating from samples with higher weights than those with lower statistical weights. Such approach has been implemented in the Cascading Adaptive Transitional Metropolis In Parallel (CATMIP) algorithm [230], a variant of the TMCMC sampler. This allows for the Markov chains to be propagated independently of each other, allowing for the computation to be done in parallel at an ensemble level.

Furthermore, an adaptive tuning algorithm is also proposed to automatically tune the step-size parameter of the AIES for every iteration. Thus, the objectives of the proposed TEMCMC sampler are to provide: 1) a sampler which is “tune-free” to the users; 2) a sampler which can be easily parallelised [231]; 3) a computationally less-expensive sampler than the existing TMCMC sampler; and 4) a robust method to effectively moderate acceptance rates within the acceptable bounds across the iterations.

The performance and the results obtained from the proposed algorithm will be compared and verified against the TMCMC. This will be done through the following numerical examples: 1) a $2DoF$ Coupled oscillator system where the objective is to infer 4 epistemic parameters; and 2) a 4-peaked Himmelblau’s function where the objective is to observe the performance of each sampler in sampling from a multi-peaked posterior. Such comparisons will be done on the basis of the estimation of the inferred parameters, computational time elapsed in generating the posterior samples, and acceptance rates across the iterations j by the respective samplers.

4.2 Limitations of current approach

A significant advantage of the TMCMC sampler is in its ability to compute the evidence term $P(\boldsymbol{D}|M)$ of the posterior distribution [51, 169]. The metric $P(\boldsymbol{D}|M)$ quantifies how well a given model M describes the available observations $\boldsymbol{D}$. The $P(\boldsymbol{D}|M)$ is estimated as the product of the mean of the nominal weights $w_i^j = P(\boldsymbol{D}|\theta_i, M)^{\Delta\beta_j}$ at any given iteration j:

$$P(\boldsymbol{D}|M) \approx \prod_{j=1}^{m} \frac{1}{N} \cdot \sum_{i=1}^{N} w_i^j \tag{4.1}$$

One key problem in the TMCMC technique is that there is no fixed universal value for the scaling parameter γ (see Eq. (3.8)) although it was stated in [51] that the “optimal” value would be 0.2. This was highlighted in [169] where it was argued that this “optimal” value is not applicable for all cases. In fact, different values for γ have been utilised in different research such as 0.5 in [174] and 1.0 in [171]. Hence, [169] proposed a tuning algorithm to adaptively adjust γ, starting from an initial value of $\gamma^{j=1} = \frac{2.4}{\sqrt{N_d}}$ [102, 232] where N_d is the dimension of $\boldsymbol{\theta}$. Upon the conclusion of the MCMC step in updating the samples, the mean acceptance rate for the current iteration α^j is obtained. This mean acceptance rate is then compared against the target acceptance rate α_{tr} which is defined as [150]:

$$\alpha_{tr} = \frac{0.21}{N_d} + 0.23 \tag{4.2}$$

Once this is done, the scale parameter is then tuned and updated according to:

$$\gamma^{j+1} = \gamma^{j} \cdot \exp\left[\frac{\alpha^{j} - \alpha_{tr}}{j}\right] \tag{4.3}$$

Such approach should also help to moderate the overall acceptance rate of the samples such that α^j falls between 0.2 and 0.5 as much as possible [150]. However, to the best of knowledge, investigations into the acceptance rates of the TMCMC sampler across iterations have not been done previously. The TMCMC sampling procedure is summarised in Algorithm 5.

Algorithm 5 TMCMC sampler algorithm

```
procedure (Generate $N$ samples from $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$)
    Set $j = 0$ and $\beta_j = 0$                                   ▷ Initialise iteration counters
    Set $\gamma^{j+1} = \frac{2.4}{\sqrt{N_d}}$                      ▷ Initialise scale parameter
    Draw initial $N$ sample set: $\boldsymbol{\theta}_i \sim P(\boldsymbol{\theta}|M)$   ▷ Generate samples from the prior
    while $\beta_j < 1$ do                                          ▷ Main sampling loop
        Set $j = j + 1$
        Compute $\Delta\beta_j$ using Eq. (3.11)
        Compute $P^j$ using Eq. (3.6)
        for $i = 1 : N$ do                                          ▷ MCMC step
            Select the $k^{th}$ Markov chain $\sim \hat{w}_k^j$
            Draw candidate sample: $\boldsymbol{\theta}_i^* \sim q(\boldsymbol{\theta}_i^*|\boldsymbol{\theta}_k)$
            Accept/Reject $\boldsymbol{\theta}_i^*$ with probability $\alpha_i$ using Eq. (3.3)
        end for
        Compute $\gamma^{j+1}$ using Eq. (4.3)
    end while
    Compute $P(\boldsymbol{D}|M)$ using Eq. (4.1)
end procedure
```

The current TMCMC sampler is also computationally inefficient since the mean $\bar{\boldsymbol{\theta}}^j$ and covariance matrix $\boldsymbol{\Sigma}^j$ need to be calculated at each iteration j (see Eq. (3.9) and (3.8) respectively). This adds additional parameters that need to be adaptively tuned, on top of β_j and γ^j, as a result of the choice of proposal distribution $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}_k)$ used. Furthermore, there is a loose assumption that it is optimal to set $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}_k)$ as a Normal distribution [50]. Strictly speaking, such choice only provides convenience in simplifying the sampling process of $\boldsymbol{\theta}^*$ from $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$ and the computation of the acceptance ratio α. Such convenience is attributed to the Normal distribution being symmetric about its mean [50, 233]. In fact, there are specific cases whereby asymmetric proposal distributions are required. For instance, if one is to estimate the posterior of a variance parameter, to ensure that the proposed candidate samples $\boldsymbol{\theta}^*$ are never less than 0, the proposal distribution should be such that it is skewed towards positive values (i.e. $\boldsymbol{\theta}^* > 0$) such as the Log-normal distribution [234]. This would ensure that the overall acceptance rates α would not be too low (i.e. $\alpha < 0.15$) [149]. Optimally, the proposal distribution should follow that of the posterior [233] but such approach is not feasible due to the lack of apriori knowledge over the analytical form of the actual posterior itself. Thus, the choice of $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}_k)$ adds some degree of uncertainty.

4.3 Transitional Ensemble Markov Chain Monte Carlo

Numerous alternatives of the MCMC kernels within the TMCMC have been considered in recent studies including: 1) Slice sampler (i.e. TMCMC-Slice) [235, 236]; 2) Metropolis-Adjusted-Langevin (MAL) sampler (i.e. L-TMCMC) [237]; and 3) Differential evolution MCMC kernel (i.e. DE-TMCMC) [238]. A summary is provided for each variant of the TMCMC sampler listed above.

The TMCMC-Slice is able to produce estimates with significantly lower COV than that of TMCMC. This is because the Slice sampler algorithm [141] draws samples more efficiently than MH from P^j such that candidate samples $\boldsymbol{\theta}_i^*$ are now drawn closer to regions of higher probability in $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$. This leads to a higher convergence of $\boldsymbol{\theta}_i$ and a smaller spread about the sample space defined by $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$ [235]. In addition, the use of the Slice sampler algorithm removes the need for $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}_i)$ thereby removing the need to compute $\bar{\boldsymbol{\theta}}^j$ and $\boldsymbol{\Sigma}^j$ [141, 236]. A significant drawback of the TMCMC-Slice is its relatively long computational time compared to the TMCMC due to the computation cost involved in tuning the auxiliary variables and step-size of the Slice sampler [236].

The L-TMCMC demonstrated high parallel efficiency by adopting an adaptive kriging metamodel, in place of the true model, to perform the model evaluations when computing the likelihood function [237]. This reduces computation time, thereby making the algorithm computationally less-expensive compared to the TMCMC [237]. In addition, due to the use of the MAL move kernel, it allows for the L-TMCMC to sample more efficiently from multi-modal $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$ as well as a better identification of parameters, especially those which were unidentifiable by TMCMC [237].

The DE-TMCMC demonstrated high parallel efficiency due to the use of the Differential evolution kernel, thereby making it computationally more efficient compared to the TMCMC [238]. In addition, the sampler is able to produce estimates with higher degree of accuracy due to its maximum a posteri estimates being closer to the true values of the inferred parameter(s) compared to the TMCMC [238]. However, the sampler still has significant number of auxiliary parameters to tune for the differential evolution MCMC kernel. This could add significant computational costs [238].

In this work, the AIES is implemented as the alternative MCMC kernel to the MH. The reasons for this are the following: 1) to exploit the efficiency of the AIES in its ability to sample from highly-skewed and anisotropic distributions; 2) the AIES can be parallelised; and 3) no proposal distribution is required which, in turn, reduces the number of auxiliary parameters to tune [5, 231]. This gives rise to the TEMCMC to which an additional feature proposed is the adaptive tuning algorithm which automatically tunes its step-size parameter and moderate the sampler's acceptance rates.

4.3.1 Review of the Affine-invariant Ensemble Sampler

The AIES is a MCMC sampling technique recently developed which possesses the affine-invariance property [5]. Currently, it has been applied across numerous research fields such as Cosmology [231, 239, 240, 241, 242, 243], Physics [244, 245, 246], and Engineering [247, 248, 249, 250, 251]. To provide an understanding of the AIES sampler, this section will first present the concept of affine-transformation, followed by a definition and description of an ensemble before finally, explaining the sampler.

An affine-transformation operation ψ is an invertible linear mapping from a $\mathbb{R}^{N_d}$ to $\mathbb{R}^{N_d}$ space [252]:

$$\psi(\boldsymbol{\theta}) : \boldsymbol{\Theta} = \hat{\boldsymbol{A}}\,\boldsymbol{\theta} + \boldsymbol{b} \tag{4.4}$$

where $\boldsymbol{\Theta}$ represents $\boldsymbol{\theta}$ in the affine-transformed space, $\hat{\boldsymbol{A}}$ is the N_d-by-N_d non-singular transformation matrix and $\boldsymbol{b}$ is the N_d-by-1 translation vector. This simple transformation transforms a difficult sampling problem into a tractable one, in a similar way in which a deterministic optimization problem is transformed to deal with a well-scaled function [5]. Let $P'(\boldsymbol{\Theta}|\boldsymbol{D}, M)$ represents a general class of densities describing the posterior distribution of $\boldsymbol{\Theta}$ where P' denotes the distribution function in the $\boldsymbol{\Theta}$-space. If the following condition holds [5]:

$$P'(\boldsymbol{\Theta}|\boldsymbol{D}, M) = P'(\psi(\boldsymbol{\theta})|\boldsymbol{D}, M) \propto P(\boldsymbol{\theta}|\boldsymbol{D}, M) \tag{4.5}$$

then, $\boldsymbol{\Theta}$-space and $\boldsymbol{\theta}$-space are said to be affine-invariant [253]. Therefore, the Affine-invariant sampler can be constructed by using a proposal distribution q' with the form [5]:

$$q'(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}_i) = q'(\psi(\boldsymbol{\theta}^*)|\psi(\boldsymbol{\theta}_i)) \propto q(\boldsymbol{\theta}^*|\boldsymbol{\theta}_i) \tag{4.6}$$

This proposal distribution is also invariant under affine-transformation. In essence, the probability of generating a sample $\boldsymbol{\Theta}^*$ given $\boldsymbol{\Theta}_i$ in the transformed $\boldsymbol{\Theta}$-space is equal to the probability of generating a sample $\boldsymbol{\theta}^*$ given $\boldsymbol{\theta}_i$ in the original $\boldsymbol{\theta}$-space [5].

Within the AIES, a collection of N_c Markov chains: $\vec{\boldsymbol{\theta}}_i = (\boldsymbol{\theta}_{1,i}, \boldsymbol{\theta}_{2,i}, \dots, \boldsymbol{\theta}_{N_c-1,i}, \boldsymbol{\theta}_{N_c,i})$ is first generated. The proposal sample $\boldsymbol{\theta}^*_k$ is then obtained using a move kernel that is invariant towards an affine-transformation and that uses the current sample and that obtained from a complementary chain. In practice, it is required that the number of chains N_c be at least twice the dimension of $\boldsymbol{\theta}$ (i.e. $N_c \geq 2 \times N_d$) [5]. Each chain generates 1 sample from the prior giving the first ensemble $\vec{\boldsymbol{\theta}}_i$ for $i = 1$. Once this is done, the samples in the ensemble are then updated one chain at a time. To update the k^{th} chain (for $k = 1, \dots, N_c$), a sample from a complementary chain is randomly selected from the set $\vec{\boldsymbol{\theta}}_{[k],i} = (\boldsymbol{\theta}_{1,i+1}, \dots, \boldsymbol{\theta}_{k-1,i+1}, \boldsymbol{\theta}_{k+1,i}, \dots, \boldsymbol{\theta}_{N_c,i})$. Let this chosen sample from the set $\vec{\boldsymbol{\theta}}_{[k],i}$ be denoted as $\boldsymbol{\theta}_{[k]}$. Following which, the candidate sample for the k^{th} chain $\boldsymbol{\theta}^*_k$ is generated. This can be expressed as [5, 231]:

$$\boldsymbol{\theta}^*_k = \boldsymbol{\theta}_{[k]} + \lambda \cdot (\boldsymbol{\theta}_{k,i} - \boldsymbol{\theta}_{[k]}) \tag{4.7}$$

whereby λ is real-valued scalar proposal stretch factor of the stretch-move affine-transformation [5, 231]. λ can be represented as a random variable following a proposal distribution $g(\lambda)$ [5]. Analogous to the MH algorithm where a symmetric $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}_i)$ is used, $g(\lambda)$ is chosen to satisfy the symmetry condition such that [5]:

$$g\left(\frac{1}{\lambda}\right) = \lambda \cdot g(\lambda) \tag{4.8}$$

so that the stretch-move in Eq. (4.7) is symmetric [5]. For this reason, $g(\lambda)$ is proposed in [5] as:

$$g(\lambda) = \begin{cases} \frac{1}{2\cdot(\sqrt{u} - \frac{1}{\sqrt{u}})} \cdot \frac{1}{\sqrt{\lambda}} & \text{if } \lambda \in [\frac{1}{u}, u] \\ 0 & \text{otherwise} \end{cases} \tag{4.9}$$

whereby u serves as the user-defined step-size of the AIES sampler which needs to be strictly greater than 1. Once $\boldsymbol{\theta}_k^*$ is sampled following Eq. (4.7), it is then accepted with probability α_k:

$$\alpha_k = \min\left[1,\ \lambda^{N_d - 1} \cdot \frac{P(\boldsymbol{\theta}_k^*|\boldsymbol{D}, M)}{P(\boldsymbol{\theta}_{k,i}|\boldsymbol{D}, M)}\right] \tag{4.10}$$

Once the samples in all N_c chains have been updated, set $i = i+1$ and the updating procedure repeats itself until $i = N$. In summary, the entire sampling procedure by AIES is summarised and illustrated in Algorithm 6. Interested readers will find further theoretical and numerical investigations of ensemble samplers with affine-invariance properties in reference [5, 101].

Figure 4.1 illustrates the stretch-move in a $2D$ sample space $\boldsymbol{\theta} = (\theta^1, \theta^2)$. In the figure, the sample of the 3^{rd} chain (in red) is being updated. The complementary sample for this case is $\boldsymbol{\theta}_{4,i}$. A straight line initiating from $\boldsymbol{\theta}_{4,i}$ is drawn to include $\boldsymbol{\theta}_{3,i}$. This straight line is the path along which the candidate sample $\boldsymbol{\theta}_3^*$ could possibly lie. By sampling λ from $g(\lambda)$, and using Eq. (4.7), $\boldsymbol{\theta}_3^*$ is defined and represented in blue.

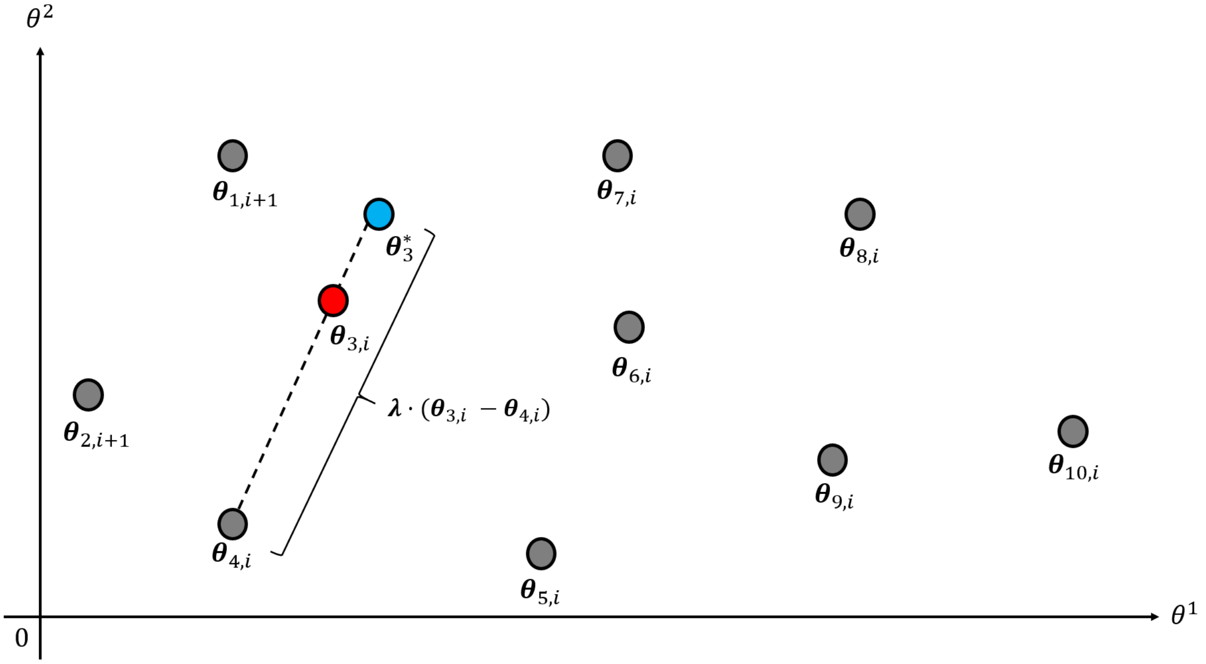

θ^2
$\boldsymbol{\theta}_{1,i+1}$
$\boldsymbol{\theta}_{7,i}$
$\boldsymbol{\theta}_3^*$
$\boldsymbol{\theta}_{8,i}$
$\boldsymbol{\theta}_{3,i}$
$\boldsymbol{\theta}_{6,i}$
$\boldsymbol{\lambda} \cdot (\boldsymbol{\theta}_{3,i} - \boldsymbol{\theta}_{4,i})$
$\boldsymbol{\theta}_{2,i+1}$
$\boldsymbol{\theta}_{10,i}$
$\boldsymbol{\theta}_{9,i}$
$\boldsymbol{\theta}_{4,i}$
$\boldsymbol{\theta}_{5,i}$
0
θ^1

Figure 4.1: Schematic diagram of the stretch-move that is used to update the sample of the 3^{rd} chain in red. Here, the candidate sample for the 3^{rd} chain is represented in blue while the randomly chosen complementary sample in this case is that from the 4^{th} chain. Image adapted from [5].

Algorithm 6 AIES sampler algorithm

```
procedure (Generate $N$ samples from $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$)
    Define $N_c$ chains: $\vec{\boldsymbol{\theta}}_1 = (\boldsymbol{\theta}_{1,1}, \boldsymbol{\theta}_{2,1}, ..., \boldsymbol{\theta}_{N_c-1,1}, \boldsymbol{\theta}_{N_c,1})$
    for $i = 1 : N - 1$ do
        for $k = 1 : N_c$ do
            Select randomly $\boldsymbol{\theta}_{[k]}$ from set $\vec{\boldsymbol{\theta}}_{[k],i}$
            Sample: $\lambda \sim g(\lambda)$
            Generate $\boldsymbol{\theta}^*_k$ using Eq. (4.7)
            Calculate acceptance probability $\alpha_{AIES}$ using Eq. (4.10)
            Sample: $r \sim U[0, 1]$
            if $\alpha_k > r$ then
                Set $\boldsymbol{\theta}_{k,i+1} = \boldsymbol{\theta}^*_k$
            else
                Set $\boldsymbol{\theta}_{k,i+1} = \boldsymbol{\theta}_{k,i}$
            end if
        end for
    end for
end procedure
```

The affine-invariant property of the stretch-move is explained as such. Supposed 2 independent sampling procedures are conducted by the AIES: one to sample $\boldsymbol{\theta}$ from $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$, and the other to sample $\boldsymbol{\Theta}$ from $P'(\boldsymbol{\Theta}|\boldsymbol{D}, M)$. Given the same sequence of λ_i in both runs for $i \geq 2$, and that the starting samples $\boldsymbol{\Theta}_1$ and $\boldsymbol{\theta}_1$ are related according to Eq. (4.10), the AIES is able to generate sample sequences such that the relationship between $\boldsymbol{\Theta}_i$ and $\boldsymbol{\theta}_i$ is always upheld for all iterations i. The mathematical illustration of the affine-invariant stretch-move is provided by Eq. (4.7). By performing an affine-transformation on both sides of the equation, one obtains [5]:

$$\boldsymbol{\Theta}^*_k : \psi(\boldsymbol{\theta}^*_k) = \psi(\boldsymbol{\theta}_{[k]}) + \lambda_i \cdot (\psi(\boldsymbol{\theta}_{k,i}) - \psi(\boldsymbol{\theta}_{[k]})) \tag{4.11}$$

Expanding and re-arranging the above equation, one will obtain the final expression [5]:

$$\boldsymbol{\Theta}_{[k]} + \lambda_i \cdot (\boldsymbol{\Theta}_{k,i} - \boldsymbol{\Theta}_{[k]}) = \hat{\boldsymbol{A}} \left[\boldsymbol{\theta}_{[k]} + \lambda_i \cdot (\boldsymbol{\theta}_{k,i} - \boldsymbol{\theta}_{[k]})\right] + \boldsymbol{b} \tag{4.12}$$

Eq. (4.12) implies 2 key things: 1) the generation of samples from the affine-transformed $\boldsymbol{\Theta}$-space is no different from sampling from the original $\boldsymbol{\theta}$-space up to an affine-transformation [5]; and 2) the probability of sampling $\boldsymbol{\Theta}^*_k$ starting from $\boldsymbol{\Theta}_{k,i}$ in the $\boldsymbol{\Theta}$-space is equal to the probability of sampling $\boldsymbol{\theta}^*_k$ starting from $\boldsymbol{\theta}_{k,i}$ in the original $\boldsymbol{\theta}$-space. The second point satisfies the condition defined by Eq. (4.6) which endows the stretch-move its affine-invariant property.

Hence, one key advantage of the AIES sampler over the MH sampler is its ability to sample from a poorly-scaled and highly-anisotropic distributions just as effectively and efficiently as it would from a well-scaled affine-transformed distribution [5, 254]. To illustrate this, a numerical study is undertaken and presented as follows:

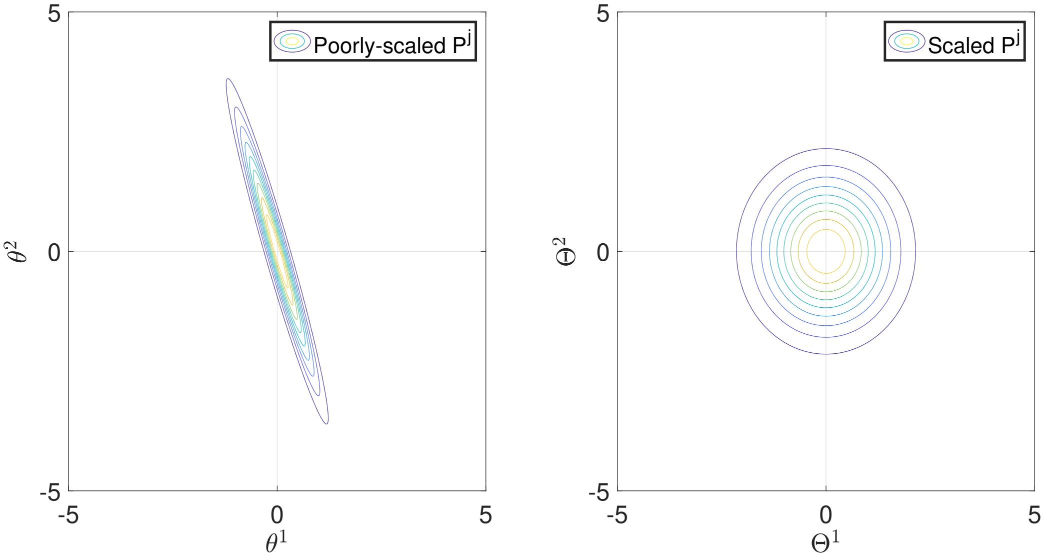


Figure 4.2: Contour plots illustrating the skewed P^j defined by Eq. (4.13) (left) and the scaled, isotropic P^j in the affine-transformed space defined by Eq. (4.15) (right).

Consider a skewed "transition" distribution defined in a $2D$ sample space $\boldsymbol{\theta} = (\theta^1, \theta^2)$ is presented in Figure 4.2 with the following mathematical expression:

$$P^j(\boldsymbol{\theta}) \propto \left\{\exp\left[-\frac{(3 \cdot \theta^1 + \theta^2)^2}{0.08} - \frac{(\theta^1 - \theta^2)^2}{2}\right]\right\}^{\beta_j} \tag{4.13}$$

where $\beta_j = 0.2$. To simplify the distribution such that it becomes easier to generate samples from, one could re-scale the problem via the following affine-transformation:

$$\begin{bmatrix} \Theta^1 \\ \Theta^2 \end{bmatrix} = \begin{bmatrix} 15 \cdot \sqrt{\beta_j} & 5 \cdot \sqrt{\beta_j} \\ \sqrt{\beta_j} & -\sqrt{\beta_j} \end{bmatrix} \begin{bmatrix} \theta^1 \\ \theta^2 \end{bmatrix} \tag{4.14}$$

This yields a relatively simpler isotropic distribution $P'^j(\boldsymbol{\Theta})$:

$$P'^j(\boldsymbol{\Theta}) \propto \exp\left[-\frac{(\Theta^1)^2}{2} - \frac{(\Theta^2)^2}{2}\right] \tag{4.15}$$

From both $P^j(\boldsymbol{\theta})$ and $P'^j(\boldsymbol{\Theta})$, 1000 samples are obtained across 4 chains (i.e. 250 samples per chain) using AIES and MH samplers. The tuning-parameter settings for the respective samplers are presented in Table 4.1 which ensures that the acceptance rates for both samples are as close to 0.234 as possible [149]. Following which, the samples in the $\boldsymbol{\Theta}$-space obtained by the respective samplers would be re-scaled to the $\boldsymbol{\theta}$-space via the inverse of Eq. (4.14). This yields the results illustrated in Figure 4.3 where it can be seen that the ECDF obtained directly from $P^j(\boldsymbol{\theta})$ and that re-scaled from $P'^j(\boldsymbol{\Theta})$ are in very good agreement. This is quantified by the area enclosed by both ECDFs where it can be seen from Table 4.2 that this area is small (i.e. close to 0) compared to that for the case of the MH sampler. In addition, it can also be observed from Figure 4.3 that the profile of the ECDF obtained directly from $P^j(\boldsymbol{\theta})$ by MH sampler (i.e. in purple) deviates significantly from that of the analytical CDF of θ^1 and θ^2. In the case of the AIES, such deviation is less significant in both dimensions. These results highlight not only the capability of the AIES in sampling directly from a skewed distribution without the need to re-scale such distribution under an affine-transformation, but also its affine-invariant property which allows it to sample from such distribution as effectively as it would from a scaled isotropic distribution. These characteristics are not exhibited by the MH sampler.

Tuning-parameter	Case: $P^j(\boldsymbol{\theta})$	Case: $P'^j(\boldsymbol{\Theta})$
Step-size, u (AIES)	8.0	8.0
Covariance matrix, $\boldsymbol{\Sigma}$ (MH)	$0.5 \cdot \boldsymbol{I}$	$5 \cdot \boldsymbol{I}$

Table 4.1: Parameter settings implemented for the respective samplers in sampling from $P^j(\boldsymbol{\theta})$ and $P'^j(\boldsymbol{\Theta})$ respectively. $\boldsymbol{I}$ denotes the identity matrix.

	AIES	MH
θ^1	0.049	0.265
θ^2	0.089	0.813

Table 4.2: Results of the area enclosed by the ECDF obtained directly from $P^j(\boldsymbol{\theta})$ and that re-scaled from $P'^j(\boldsymbol{\Theta})$ for the respective samplers.

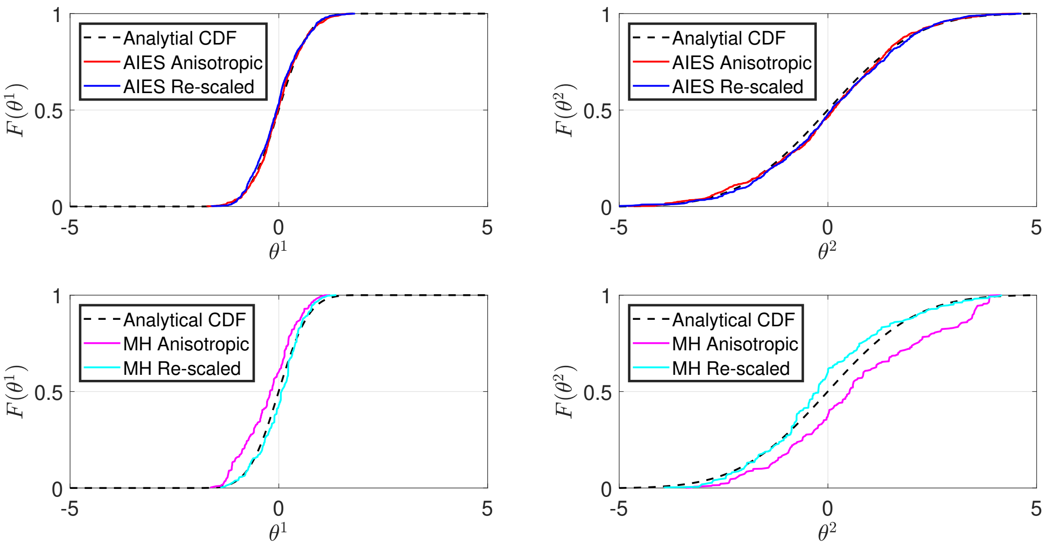


Figure 4.3: ECDFs of θ^1 and θ^2 obtained directly from $P^j(\boldsymbol{\theta})$ and those re-scaled from $P'^j(\boldsymbol{\Theta})$ when using MH and AIES.

Another advantage that the AIES sampler has over the MH sampler is the reduced number of parameters to tune adaptively given that the sample mean $\bar{\boldsymbol{\theta}}^j$ and its covariance matrix $\boldsymbol{\Sigma}^j$ need not be computed at every j. This leaves β_j and u as the only parameters to be adaptively tuned, thereby improving the computational efficiency for the TEMCMC sampler.

As such, this motivates the implementation of the AIES algorithm in the proposed TEMCMC sampler given that the "transitional" distributions P^j (see Eq. (3.6)) can be highly-skewed and anisotropic in general and its sampling performance would be least affected by the scaling of P^j across the transition step j.

4.3.2 Sample updating strategy

In both the TMCMC and the proposed TEMCMC samplers, the samples $\boldsymbol{\theta}_i$ are updated according to P^j for a given iteration j through their respective MCMC kernel. This allows for samples obtained from previous iteration with lower statistical weights $\hat{w}_i^j$ to be replaced with those with higher values of $\hat{w}_i^j$ in the current iteration j [51, 230].

However, the main difference in the sample updating strategy between the TMCMC and the TEMCMC samplers is as follows: For the case of the TMCMC sampler, samples with higher $\hat{w}_i^j$ at any given iteration j are assembled into longer Markov chains than those with lower $\hat{w}_i^j$ as described in Chapter 3. On the other hand, for the case of the TEMCMC sampler, upon computing $\hat{w}_i^j$ of the samples $\boldsymbol{\theta}_i$ obtained from previous iteration, the initiating sample of the Markov chain is obtained through weighted random sampling of $\boldsymbol{\theta}_i$ according to $\hat{w}_i^j$. This results in more Markov chains initiating from $\boldsymbol{\theta}_i$ with higher $\hat{w}_i^j$, resulting in an increased concentration of Markov chains in regions of P^j with higher probabilities. As such, the sample updating strategy adopted by the TEMCMC sampler works in similar fashion to the Neighbourhood algorithm [187] which explores the sample space by concentrating the random walk sampling in regions of the sample space of the target distribution with higher probabilities [230].

Due to the sample updating strategy adopted by the TEMCMC algorithm, this allows the Markov chains to be propagated independently of one another, allowing for the procedure to be easily parallelised [230].

4.3.3 Adaptive-tuning algorithm

To adaptively tune the step-size parameter u, an algorithm is proposed based on the work by [169]. The initial step-size value $u^{j=1}$ is set at 2 given that this is the "optimal" value for most problems [231, 239]. From this initial value, the nominal step-size u_{nom} is computed after the MCMC step:

$$u_{nom} = u^j \cdot \exp\left[\alpha^j - \alpha_{tr}\right] \tag{4.16}$$

where α^j is the acceptance rate for the current iteration and α_{tr} is the target acceptance rate (see Eq. (4.2)). The acceptance rate α^j is treated as a random variable. Consequently, the nominal step-size u_{nom} is also a random variable which can be randomized and adapted through α^j. Such adaptivity is not limited to the AIES, but is a generic property of MCMC samplers [169, 255] subjected to the Diminishing Adaptation and Bounded Convergence conditions defined in [256]. If $u_{nom} > 1$, then $u^{j+1} = u_{nom}$. Otherwise, the algorithm sets $u^{j+1} = 1.01$ to ensure that the step-size would never fall below 1. This procedure is repeated at the end of every iteration until the last transition step $j = m$.

In summary, the proposed TEMCMC sampler possesses 3 key benefits: 1) it is practically "tune-free" for the users; 2) it is computationally less expensive compared to TMCMC sampling; and 3) its acceptance rate is moderated such that it falls within the acceptance range between 0.15 and 0.50 [149] for the majority of the transition steps j. A pseudoalgorithm of the TEMCMC sampler is provided in Algorithm 7. To illustrate such benefits and strengths of the TEMCMC sampler, the proposed algorithm will be implemented in 2 numerical examples: 1) the $2DoF$ Coupled oscillator system; and 2) the $2D$ Multi-modal posterior. Within these numerical examples, the sampling performance and estimation results obtained by the TEMCMC sampler will be compared against the TMCMC sampler due to the latter being the state-of-the-art sampling algorithm at the time the proposed sampler was conceptualised.

Algorithm 7 Proposed TEMCMC sampler algorithm

```
procedure (Generate $N$ samples from $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$)
    Set $j = 0$ and $\beta_j = 0$                                    ▷ Initialise
    Draw $N$ initial sample set: $\boldsymbol{\theta}_i \sim P(\boldsymbol{\theta}|M)$
    Set $a^{j+1} = 2$                                                ▷ Set initial value of step-size
    while $\beta_j < 1$ do                                           ▷ Main sampling loop
        Set $j = j + 1$
        Compute $\Delta\beta_j$ using Eq. (3.11)
        Compute $P^j$ using Eq. (3.6)
        Resample $N$ samples: $\boldsymbol{\theta}_i \sim \hat{w}_i^j$
        Set $\boldsymbol{\theta}_i = \boldsymbol{\theta}_{i,1}$ in ensemble $\vec{\boldsymbol{\theta}}_1$          ▷ Initiate ensemble
        Update $\vec{\boldsymbol{\theta}}_1$ with 1 iteration of AIES (see Algorithm 6)   ▷ MCMC step
        Set samples in ensemble $\vec{\boldsymbol{\theta}}_2$ as samples of $P^j$
        Compute $u_{nom}$ using Eq. (4.16)                           ▷ Tuning the step-size
        if $a_{nom} > 1$ then
            Set $u^{j+1} = u_{nom}$
        else
            Set $u^{j+1} = 1.01$
        end if
    end while
    Compute $P(\boldsymbol{D}|M)$ using Eq. (4.1)
end procedure
```

4.4 Numerical example 1: 2DoF Coupled oscillator system

The objective of this numerical example is to observe and compare the differences in the performance of the TMCMC sampler and the proposed TEMCMC sampler in a 4-dimensional Bayesian model updating set-up. This comparison will be done on the basis of: 1) the computation time elapsed in sampling from the posterior; 2) the transition step size $\Delta\beta_j$

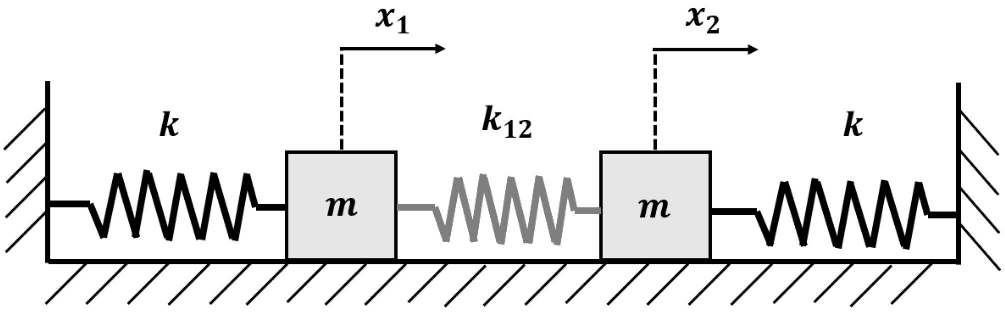


Figure 4.4: Schematic diagram of the $2DoF$ Coupled oscillator system based on the set-up in [6].

across sampling iterations j; 3) the acceptance rates across the transition steps j; and 4) the Monte Carlo standard error σ_{SE} of the estimation of the epistemic parameters $\boldsymbol{\theta}$.

Figure 4.4 illustrates a simple 2 Degrees-of-Freedom (DoF) coupled oscillator set-up consisting of 2 equal-sized blocks with equal mass m attached to primary springs with stiffness k and an inter-mass secondary spring with stiffness k_{12}. x_1 and x_2 denote the respective instantaneous displacement of the blocks. In this problem, the mass of the blocks are fixed at $m = 0.5\ kg$. In addition, it is assumed that both k and k_{12} take on fixed values, $\{k, k_{12}\} = \{0.6, 1.0\}\ N/m$, but these values are not known (i.e. epistemic uncertainty). In order to infer k and k_{12}, measurements are obtained in the form of the eigenfrequencies $\boldsymbol{D} = (\omega_1, \omega_2)$ whereby ω_1 and ω_2 are the 2 eigenfrequencies associated with the in-phase and out-of-phase mode shapes of the symmetric system respectively which can be easily computed as follows [6]:

$$\hat{\omega}_1 = \sqrt{\frac{k}{m}} \tag{4.17}$$

$$\hat{\omega}_2 = \sqrt{\frac{(k + 2 \cdot k_{12})}{m}} \tag{4.18}$$

The above equations constitute the model class $M = \{\hat{\omega}_1, \hat{\omega}_2\}$ to be updated. The frequency measurements of ω_1 and ω_2 are however corrupted with "noise" ϵ_1 and ϵ_2 respectively such that:

$$\omega_1 = \hat{\omega}_1 + \epsilon_1 \tag{4.19}$$

$$\omega_2 = \hat{\omega}_2 + \epsilon_2 \tag{4.20}$$

whereby ϵ_1 and ϵ_2 are the "noise" terms following a Normal distribution with means 0 Hz

and standard deviations σ_1 and σ_2 respectively. Here, σ_1 and σ_2 are fixed values set at 10 % of the nominal values of ω_1 and ω_2 respectively. This yields $\{\sigma_1, \sigma_2\} = \{0.110, 0.228\}$ Hz. For simplicity, it is assumed that the measurement "noise" ϵ_1 and ϵ_2 are not correlated and that the "noise" between individual measurements of ω_1 and ω_2 are also independent. In this problem, 15 independent realizations of ω_1 and ω_2 are obtained and these synthetic data are presented in the form of a scatterplot shown in Figure 4.5 while the numerical values are presented in Table 4.3.

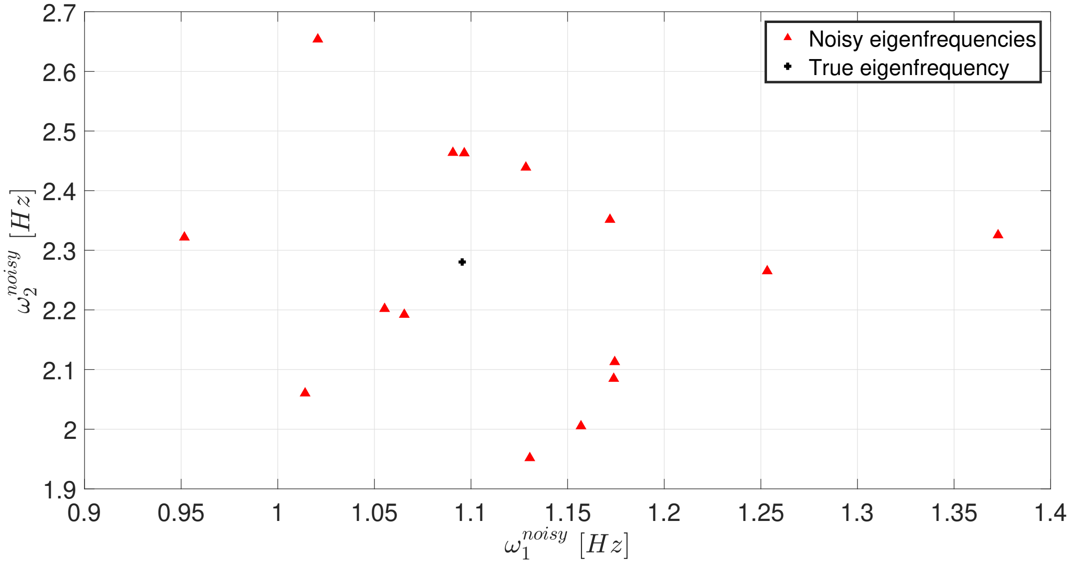


Figure 4.5: Scatterplot of the 15 different "measured" values ω_1 and ω_2.

Measurement no.	ω_1 $[Hz]$	ω_2 $[Hz]$	Measurement no.	ω_1 $[Hz]$	ω_2 $[Hz]$
1	1.172	2.351	9	1.055	2.202
2	1.097	2.463	10	1.253	2.265
3	1.157	2.005	11	0.952	2.322
4	1.091	2.464	12	1.130	1.952
5	1.021	2.654	13	1.174	2.085
6	1.373	2.325	14	1.066	2.192
7	1.174	2.113	15	1.014	2.060
8	1.128	2.439	–	–	–

Table 4.3: Numerical values of ω_1 and ω_2 shown in Figure 4.5.

4.4.1 Bayesian model updating

For this problem, the priors for k and k_{12} are set as Uniform priors taking values between 0.01 N/m and 4.0 N/m. In addition, despite σ_1 and σ_2 being predetermined values, in reality, these 2 parameters are unknown and are also set as epistemic parameters to be inferred. The priors for σ_1 and σ_2 are also set as Uniform priors taking values between 1.0×10^{-5} Hz and 1.0 Hz. Therefore, the total number of epistemic parameters to 4, thereby making this a 4-dimensional Bayesian model updating problem: $\boldsymbol{\theta} = (k, k_{12}, \sigma_1, \sigma_2)$. It is assumed that the epistemic parameters are independent from one another. The likelihood function is modelled to follow a Normal distribution and assuming independence between individual observations, it is expressed as follows:

$$P(\boldsymbol{D}|\boldsymbol{\theta}, M) = \prod_{n=1}^{15} \frac{1}{2 \cdot \pi \cdot \sigma_1 \cdot \sigma_2} \cdot \exp\left[-\frac{(\omega_{1,n} - \hat{\omega}_1)^2}{2 \cdot {\sigma_1}^2} - \frac{(\omega_{2,n} - \hat{\omega}_2)^2}{2 \cdot {\sigma_2}^2}\right] \tag{4.21}$$

4.4.2 Results and discussions

From the posterior $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$, $N = 1000$ samples are generated using the TMCMC sampler and the proposed TEMCMC sampler and the sampling time elapsed is of 48.60 s and 6.52 s respectively. The reason behind the significantly large difference in the sampling time lies in the different sample updating strategy (i.e. the MCMC step) adopted by the respective sampling algorithm. In the case of the TMCMC sampler, the sample updating procedure in the MCMC step is performed in a serial manner after each weighted resampling of $\hat{w}_i^j$ (i.e. see Algorithm 5). In the case of the TEMCMC sampler, on the other hand, the sample updating procedure is done at an ensemble level and involves only 1 iteration of the AIES procedure (i.e. see Algorithm 7). This allows for the computation time of the TEMCMC sampler to be significantly reduced.

Sampler	$\Delta\beta_1$	$\Delta\beta_2$	$\Delta\beta_3$	$\Delta\beta_4$	$\Delta\beta_5$	$\Delta\beta_6$	$\Delta\beta_7$	β_8
TMCMC	0.011	0.040	0.073	0.100	0.139	0.185	0.241	0.212
TEMCMC	0.012	0.042	0.061	0.076	0.163	0.376	0.270	–

Table 4.4: Results of $\Delta\beta_j$ computed for the respective samplers.

In sampling from $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$, the TMCMC sampler required 8 iterations whereas the TEMCMC sampler required only 7 iterations. Figure 4.6 shows the evolution of the statistics of β_j and of the acceptance rates across all iterations j for both TMCMC and TEMCMC samplers. From the figure, it can be seen that the rate of increase of β_j is generally higher

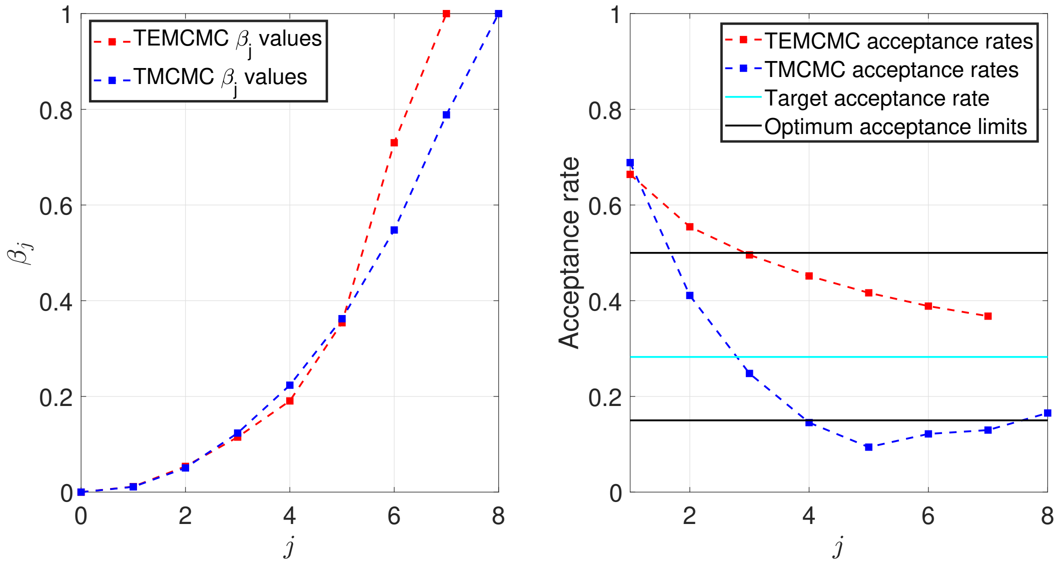


Figure 4.6: The statistics of β_j and the acceptance rates across all iterations j. The target acceptance rate is 0.283.

in the case of the TEMCMC sampler than that for the TMCMC sampler. To support such observation, the value of $\Delta\beta_j$ between successive iterations is computed using the following equation:

$$\Delta\beta_j = \beta_j - \beta_{j-1} \tag{4.22}$$

The results of $\Delta\beta_j$ for the corresponding samplers are presented in Table 4.4. From the table, it can be seen that the value of $\Delta\beta_j$ at any given j is generally higher for the case of the TEMCMC sampler compared to the TMCMC sampler. Coupled with the fact that the TEMCMC sampler required 1 less iteration compared to the TMCMC sampler in sampling from $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$, it indicates a significantly improved mixing performance by the TEMCMC sampler. This highlights the effectiveness of the AIES move kernel compared to the MH move kernel in the TMCMC sampler.

In addition, it can also be observed from Figure 4.6 that the acceptance rates for the TMCMC sampler shows a higher degree of variation compared to that of the TEMCMC sampler. In fact, only 3 out of 8 iterations yielded acceptance rate values within the optimal limits in the case of the TMCMC sampler whereas 5 out of 7 iterations yielded acceptance rate values within the optimal limits the case of the TEMCMC sampler. This highlights the effectiveness of the proposed adaptive tuning algorithm in moderating the step-size u in the TEMCMC sampler such that for majority of the iterations, the acceptance rates of the sampler is kept within the optimal range.

The resulting scatterplot matrix of the posterior samples is shown in Figure 4.7 and from there, the statistics of the estimates of the epistemic parameters by the respective samplers are obtained in the form of the sample mean estimates and the Monte Carlo standard error σ_{SE} computed from Eq. (3.4) with batch size $n_b = 100$. The results of the statistics of the estimates for the corresponding inferred parameters are presented and summarised in Table 4.5. From the table, it can be seen that the estimation statistics obtained by the TEMCMC sampler is generally similar and is well-verified against that obtained by the TMCMC sampler.

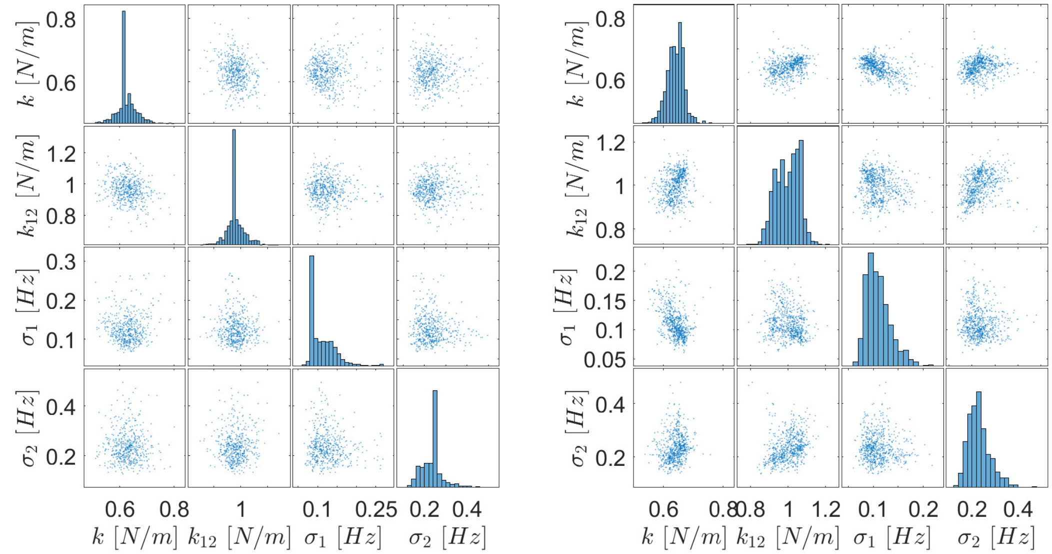


Figure 4.7: Scatterplot matrix illustrating the resulting posterior of the epistemic parameters obtained by the TMCMC sampler (left) and the TEMCMC sampler (right).

Finally, the resulting scatterplot of the model output is compared against the scatterplot of the data for ω_1 and ω_2. The results of the model updating for the case of the TMCMC sampler and the TEMCMC sampler are illustrated in Figure 4.8. As seen from the figure, the resulting model outputs from the posterior samples obtained encompass the true solution of ω_1 and ω_2 for both the TMCMC and TEMCMC samplers. This highlights that the TEMCMC sampler works just as effectively as the TMCMC sampler as a tool in Bayesian model updating.

$\boldsymbol{\theta}$	Reference value	TMCMC		TEMCMC	
		$E[\boldsymbol{\theta}]$	σ_{SE}	$E[\boldsymbol{\theta}]$	σ_{SE}
k	$0.6\ N/m$	$0.625\ N/m$	$8.51 \times 10^{-4}\ N/m$	$0.640\ N/m$	$1.10 \times 10^{-3}\ N/m$
k_{12}	$1.0\ N/m$	$0.962\ N/m$	$2.00 \times 10^{-3}\ N/m$	$0.998\ N/m$	$2.80 \times 10^{-3}\ N/m$
σ_1	$0.110\ Hz$	$0.114\ Hz$	$8.59 \times 10^{-4}\ Hz$	$0.110\ Hz$	$8.02 \times 10^{-4}\ Hz$
σ_2	$0.228\ Hz$	$0.236\ Hz$	$1.60 \times 10^{-3}\ Hz$	$0.230\ Hz$	$1.60 \times 10^{-3}\ Hz$

Table 4.5: A summary of the statistics of the estimation of the epistemic parameters $\boldsymbol{\theta} = (k, k_{12}, \sigma_1, \sigma_2)$ via the posterior samples obtained using the TMCMC and TEMCMC samplers.

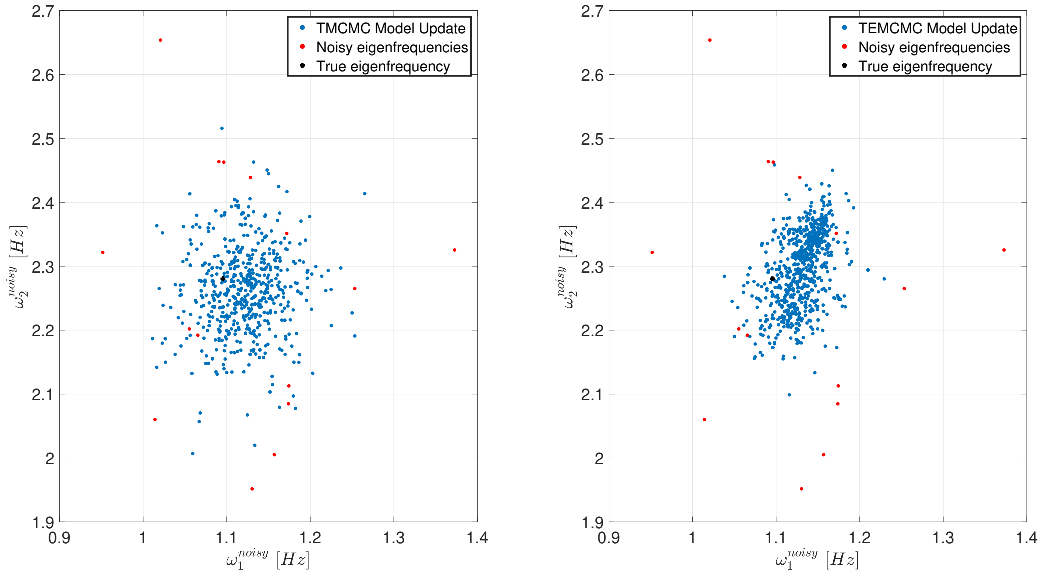


Figure 4.8: Scatterplot matrix illustrating the model updating results attained by the TMCMC and TEMCMC samplers.

4.5 Numerical example 2: 2D Multi-modal posterior

The following set-up is based on the problem presented in [257] and the objective of this numerical example is to observe and compare the differences in the performance of the TMCMC sampler and the proposed TEMCMC sampler in generating samples from a multimodal posterior. This comparison will be done on the basis of the sample distribution in relation to the analytical solution of the posterior, the computation time elapsed in sampling from the posterior, and the acceptance rates across the transition steps j.

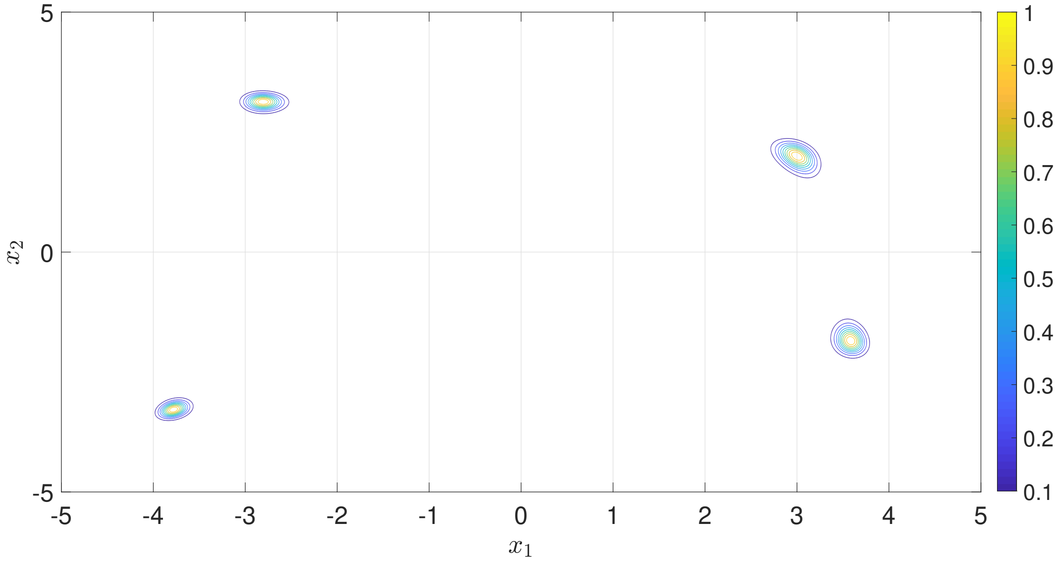


Figure 4.9: Contour plot illustration of the 4-peaked posterior based on the Himmelblau's function [7]. The numbers on the colour chart represent the height of the posterior computed from Eq. (4.24).

In this study, a 2-dimensional posterior with 4 peaks, defined by the dimensionless variables x_1 and x_2, is presented and illustrated in Figure 4.9. The analytical function of this posterior is based upon the Himmelblau's function which is a test-function used in mathematical optimisation problems to test the performance of optimisation algorithms. The Himmelblau's function $H(x_1, x_2)$ is mathematically defined as [7]:

$$H(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \tag{4.23}$$

which yields 1 solution of local maximum at $\{x_1, x_2\} = \{-0.271, -0.923\}$ and 4 distinct solutions of local minima at $\{x_1, x_2\} = \{3.0, 2.0\}$, $\{-2.805, 3.131\}$, $\{-3.779, -3.283\}$, and $\{3.584, -1.848\}$. From which, the posterior of interest is then defined as follows [257]:

$$P(\boldsymbol{\theta}|\boldsymbol{D}, M) \propto \exp\left[-H(x_1, x_2)\right] \tag{4.24}$$

which ensures that the local minimia of $H(x_1, x_2)$ now becomes the region of high probability giving rise to the 4 peaks.

4.5.1 Bayesian model updating

For this problem, the epistemic parameters are x_1 and x_2, thereby making this a 2-dimensional Bayesian model updating problem: $\boldsymbol{\theta} = (x_1, x_2)$. The priors for x_1 and x_2 are set as Uniform priors taking values between -5 and 5 and it is assumed that the epistemic parameters are independent from one another. The likelihood function is modelled as the exponential function of $-H(x_1, x_2)$ and thus takes on the same mathematical form as the posterior in Eq. (4.24).

4.5.2 Results and discussions

From the posterior $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$, $N = 1000$ samples are generated using the TMCMC sampler and the proposed TEMCMC sampler and the sampling time elapsed is of 16.14 s and 4.96 s respectively. The underlying reason behind the difference in the sampling time between the 2 algorithms is provided in Section 4.4.2.

Sampler	$\Delta\beta_1$	$\Delta\beta_2$	$\Delta\beta_3$	$\Delta\beta_4$	$\Delta\beta_5$
TMCMC	0.015	0.041	0.140	0.464	0.340
TEMCMC	0.016	0.042	0.155	0.508	0.279

Table 4.6: Results of $\Delta\beta_j$ computed for the respective samplers.

In sampling from $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$, both samplers required 5 iterations. Figure 4.10 shows the evolution of the statistics of β_j and the acceptance rates across all iterations j for both TMCMC and TEMCMC samplers. From the figure, it can be seen that while the evolution of β_j across all j are identical for both the TMCMC and TEMCMC samplers, the rate of increase of β_j is slightly greater for the case of the TEMCMC sampler than it is for the TMCMC sampler. To support such observation, the value of $\Delta\beta_j$ is computed (i.e. see Eq. (4.22)) for which the results for the corresponding samplers are presented in Table 4.6. From the table, it can be seen that the value of $\Delta\beta_j$ at any given j is generally higher for the case of the TEMCMC sampler compared to the TMCMC sampler. This indicates a better mixing performance as a result of the implementation of the AIES as the MCMC move kernel of the TEMCMC sampler compared to the MH in the case of the TMCMC sampler.

In addition, it can also be observed from Figure 4.10 that out of 5 iterations, only 2 lie within the optimal limits of acceptance rate for the case of the TMCMC sampler and 3 for the case of the TEMCMC sampler. For the case of the TEMCMC sampler, the acceptance rate values appear to tend towards the target acceptance rate value with increasing j while this is not the case for the TMCMC sampler. This indicates the effectiveness of the adaptive tuning algorithm for step-size u in the TEMCMC sampler in ensuring the majority of the iterations have acceptance rates within the optimal limits and that the tuning is done with reference to the target acceptance rate.

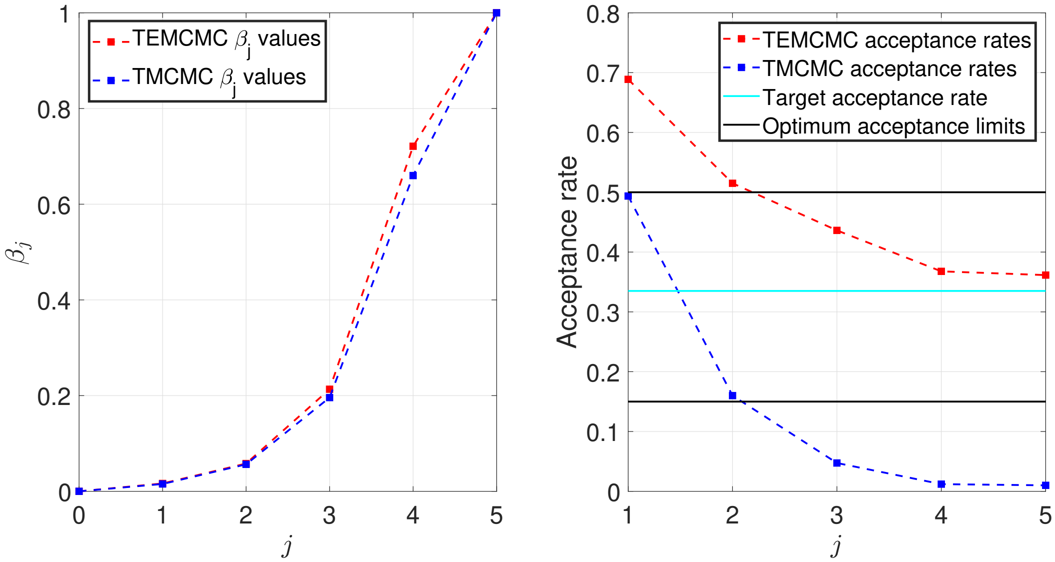


Figure 4.10: The statistics of β_j and the acceptance rates across all iterations j. The target acceptance rate is 0.335.

For the purpose of illustration, the scatterplots obtained from the transition distributions P^j between $j = 0$ and $j = 5$ via the TMCMC and TEMCMC samplers are presented in Figures 4.11 and 4.12 respectively. In both Figures 4.11 and 4.12, the scatterplots obtained by TMCMC and TEMCMC samplers are compared against the contour plot profile of the analytical solutions and it can be observed that in both figures, the scatterplot profiles match closely to the corresponding contour plot profile. This validates the effectiveness of both samplers in sampling from all P^j. Figure 4.13 presents the scatterplot profile of the samples of the final posterior $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$ in comparison with the contour plot profile of the analytical solution. While both samplers are able to sample effectively from the 4-peaked posterior, upon closer inspection, it can be observed that there is significantly less exploration of the sample space by the samples obtained via the TMCMC sampler compared to the TEMCMC sampler. This is due to the TMCMC sampler having a very low rate of acceptance in the latter iterations $j = 4$ and $j = 5$ where the acceptance rates are approximately 0.013 as seen in Figure 4.10. This reduces the number of unique samples generated from the MCMC step of the TMCMC algorithm leading to many repeated samples and a poor exploration of the sample space.

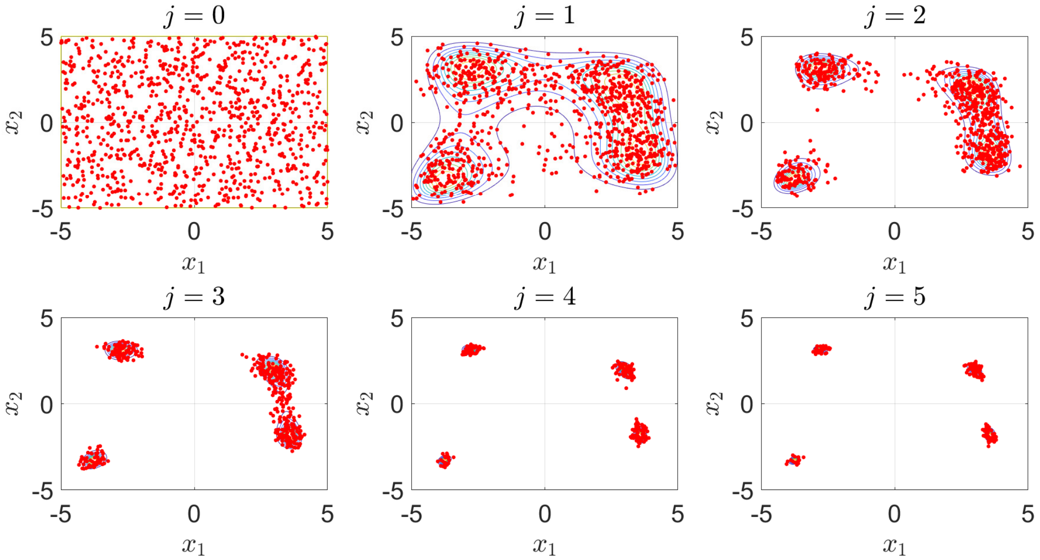


Figure 4.11: Scatterplots obtained from P^j for $j = 0$ to $j = 5$ via the TMCMC sampler.

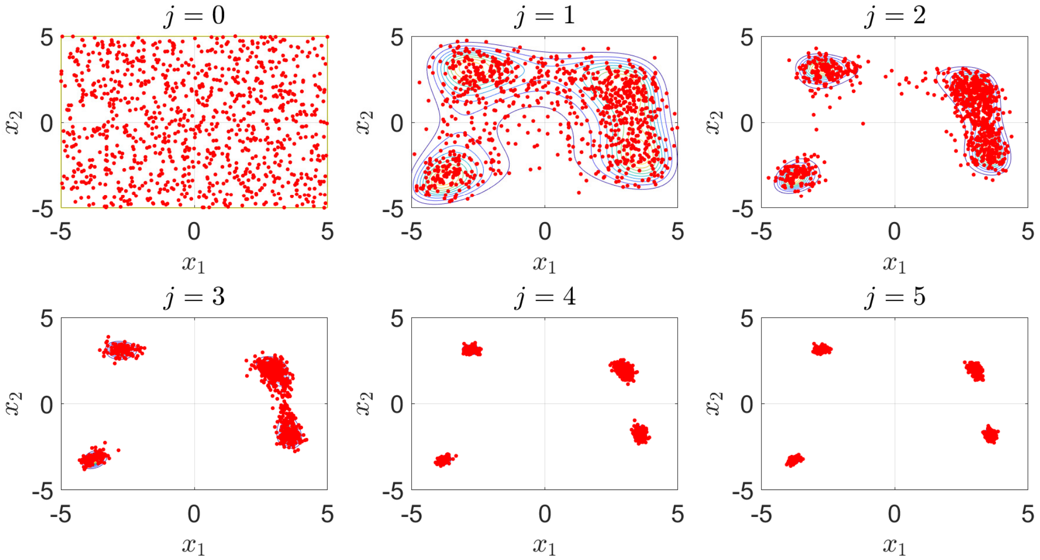


Figure 4.12: Scatterplots obtained from P^j for $j = 0$ to $j = 5$ via the TEMCMC sampler.

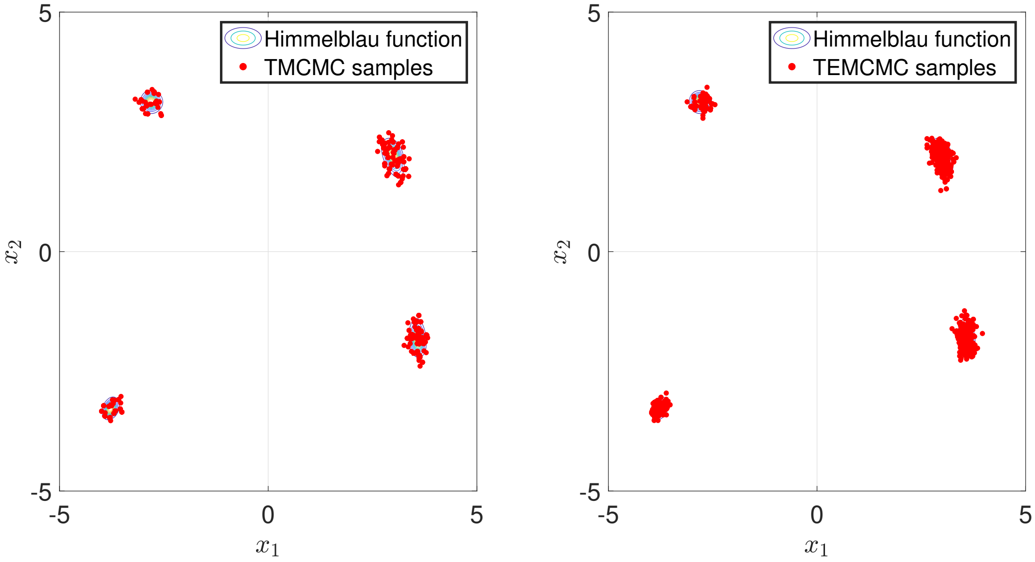


Figure 4.13: Resulting scatterplots of the samples from the final posterior $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$ obtained via TMCMC (left) and TEMCMC (right) samplers along with the analytical contour plot profile as a comparison.

4.6 Chapter conclusion

An efficient and robust sampler named Transitional Ensemble Markov Chain Monte Carlo has been proposed for Bayesian inference. The proposed sampler uses an Affine-invariant Ensemble sampler in place of the traditional Metropolis-Hasting sampler and includes an adaptive tuning algorithm making the approach "tune-free" for users. The proposed sampler out-performs the current samplers available in sampling from highly-skewed, anisotropic distributions [5, 254] such as the transition distributions by exploiting the advantages of the Affine-invariant Ensemble sampler. The absence of a need by the user to define the proposal distribution eradicates the model uncertainty in considering the choice of the distribution model and the proposal distribution used by the Affine-invariant Ensemble sampler reduces the number of parameters to tune adaptively. In addition, an efficient sample updating strategy is implemented which allows samples to be updated in parallel at an ensemble level via the Affine-invariant Ensemble sampler thereby reducing the computational cost of the proposed sampler significantly.

To illustrate the efficiency and effectiveness of the proposed Transitional Ensemble Markov Chain Monte Carlo sampler, two numerical examples have been presented showing the applicability of the approach in Bayesian model updating as well as in the presence of a complex a multi-modal posterior. In all the examples investigated, the Transitional Ensemble Markov Chain Monte Carlo sampler outperforms the traditional Transitional Markov Chain Monte Carlo sampler by requiring a shorter time whilst ensuring that the acceptance rates are well-moderated within optimal bounds. In addition to this, the transition step size by Transitional Ensemble Markov Chain Monte Carlo sampler is shown to be generally larger than that by the traditional Transitional Markov Chain Monte Carlo sampler which indicates an improved mixing performance by the Affine-invariant Ensemble sampler MCMC kernel.

In conclusion, the results from the examples presented in the chapter highlights the following key strengths of the proposed Transitional Ensemble Markov Chain Monte Carlo sampler: 1) it allows for the sampling of badly-scaled and highly-anisotropic distributions without requiring extra computational costs; 2) it is free from tuning by the user; and 3) it is more robust than the Transitional Markov Chain Monte Carlo in controlling the acceptance rates automatically. One significant drawback of this sampler, however, is the relative complexity in coding the Transitional Ensemble Markov Chain Monte Carlo algorithm from scratch compared to the standard Transitional Markov Chain Monte Carlo. To address this issue, access to the MATLAB code is provided on GitHub: `https://github.com/Adolphus8/Transitional_Ensemble_MCMC.git`

Chapter 5

Robust Off-line Bayesian Identification under Model Uncertainty

Abstract

The chapter illustrates the implementation of the Transitional Ensemble Markov Chain Monte Carlo sampler, presented in Chapter 4, in an experimental example involving a $2DoF$ Shear aluminium frame structure. For the work presented in the chapter, the positions of the 2 moveable masses within the given structure, used to simulate the damage location, are to be inferred under uncertainty over the likelihood functions. To account for such model uncertainty in estimating the mass positions, the Robust Bayesian approach is implemented. In doing so, the estimates would not only account for the uncertainty in the given data, but also the uncertainty over the choice of likelihood function used to construct the resulting posterior distribution of the mass positions. There are 2 objectives to this study: 1) to validate the proposed sampler in the Bayesian identification of the mass position parameters of the structure using actual experimental data obtained under realistic and challenging settings; and 2) to highlight the applicability of the Transitional Ensemble Markov Chain Monte Carlo sampler in the context of structural health monitoring.

5.1 Background

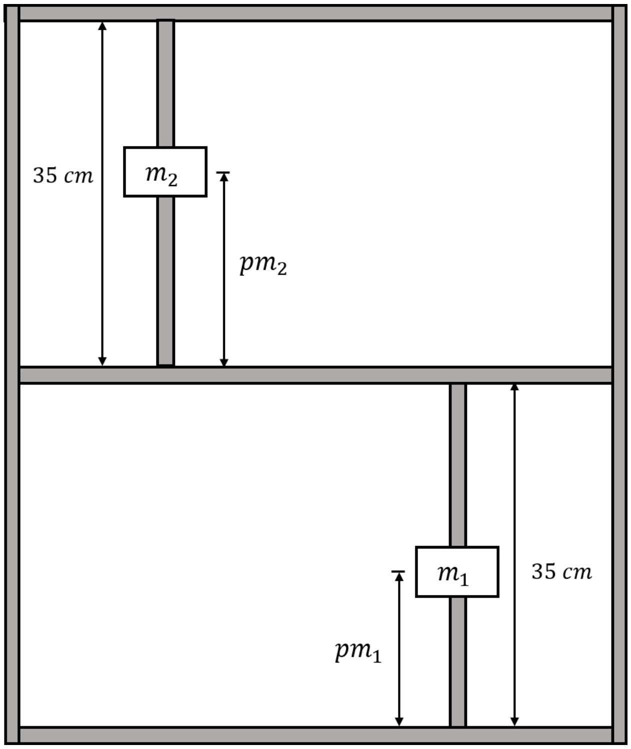


Figure 5.1: Schematic diagram of the aluminium frame with moveable masses m_1 and m_2 [1, 4, 8].

The following example involves a $2DoF$ Shear aluminium frame structure which has been presented in [1, 4, 8]. As per illustrated in Figure 5.1, the structure consists of 7 beams (3 horizontals, 2 long verticals, and 2 short verticals) and 2 movable masses m_1 and m_2 with positions pm_1 and pm_2 respectively. Recently, the structure was presented as a case study for structural health monitoring in the work by Dr. Roberto Rocchetta [1] whereby the mass positions pm_1 and pm_2 are used to simulate the structural damage location.

The experimental data was obtained via hammer impact from which the response frequencies of the structure are obtained and processed for a given combination of $\{pm_1, pm_2\}$. The response frequencies consist of the 6 natural frequencies corresponding to:

- ω_1 - the 1^{st} mode;
- ω_2 - the 1^{st} out-of-plane bending mode;
- ω_3 - the 1^{st} torsional mode;

- ω_4 - the 2^{nd} in-plane bending mode;
- ω_5 - the 2^{nd} order out-of-plane bending mode;
- ω_6 - the 2^{nd} torsional modes.

A total of 11 sets of experimental data of the response frequencies are obtained using 11 distinct combinations of $\{pm_1, pm_2\}$ in a previous experimental campaign conducted by Dr. Peng Liang [4]. These are summarised in Table 5.1.

Exp. no.	$\{pm_1, pm_2\}$	ω_1	ω_2	ω_3	ω_4	ω_5	ω_6
	$[cm]$	$[Hz]$	$[Hz]$	$[Hz]$	$[Hz]$	$[Hz]$	$[Hz]$
1	$\{5, 5\}$	20.11	22.79	47.52	63.96	183.82	283.51
2	$\{5, 20\}$	18.72	20.46	46.97	72.24	214.84	296.32
3	$\{5, 35\}$	17.715	18.29	46.42	63.45	196.38	278.70
4	$\{20, 5\}$	19.40	22.39	46.32	61.78	173.49	259.76
5	$\{20, 20\}$	17.91	20.28	45.67	64.73	190.84	284.09
6	$\{20, 35\}$	16.71	18.21	45.18	56.53	177.97	264.44
7	$\{35, 5\}$	17.71	21.76	44.00	59.48	164.05	254.48
8	$\{35, 20\}$	16.91	19.82	43.15	60.06	175.75	279.10
9	$\{35, 35\}$	15.95	17.89	42.44	50.66	163.55	257.82
10	$\{11, 11\}$	19.58	21.73	47.00	67.54	196.21	285.95
11	$\{29, 29\}$	16.65	18.85	43.93	55.43	174.35	284.84

Table 5.1: A summary of the experimental data obtained from the hammer impact test. The data is obtained from [4].

5.2 Surrogate model

For this problem, an Artificial Neural Network (ANN) is used as a surrogate model M in place of the computationally-expensive Finite Element Model (FEM) of the structure to perform Bayesian model updating and infer $\boldsymbol{\theta} = (pm_1, pm_2)$.

An ANN consists of a number of processors, known as neurons, which are analogous to the biological neuron in the brain. It defines a function $g(\boldsymbol{\theta})$ which consists of a non-linear weighted sum of other functions $h(\boldsymbol{\theta})$, thus, taking the following form:

$$\hat{\boldsymbol{\omega}} = g(\boldsymbol{\theta}) = \sum_{q} K\left(w_q \cdot h_q(\boldsymbol{\theta})\right) \tag{5.1}$$

where $\hat{\boldsymbol{\omega}} = \{\hat{\omega}_1, \ldots, \hat{\omega}_6\}$ is the vector of the model prediction of the 6 natural frequencies by the ANN, w_q is the weight to be updated in the training algorithm, and $K(\bullet)$ is the activation function to which examples include the logistic or hyperbolic functions [258]. A typical architecture of an ANN comprises of an input layer, one or more hidden layer(s), and an output layer. Within each layer, there exists multiple nodes which are connected to the nodes belonging to the next layer via weighted links.

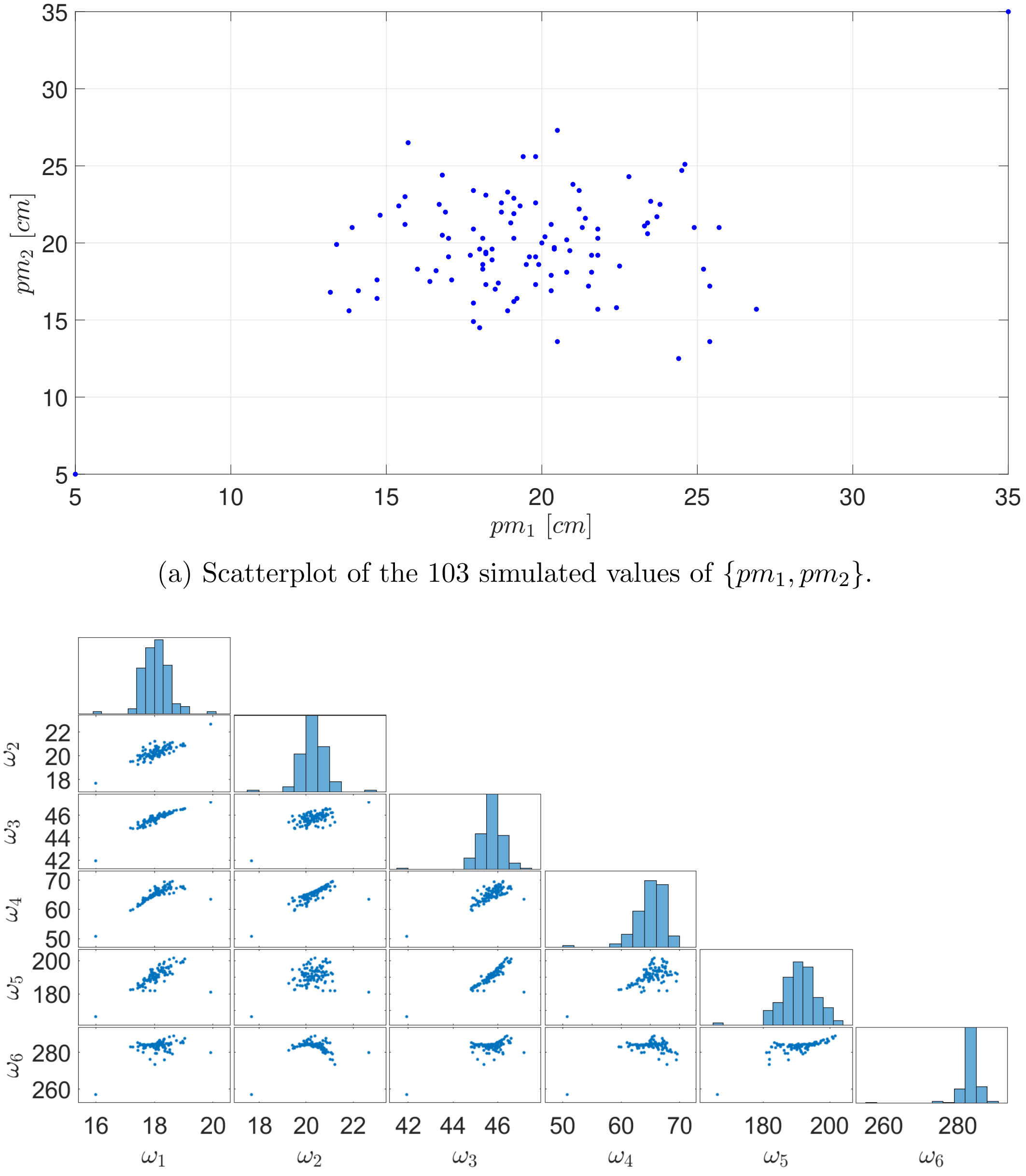
When training the ANN, the inputs defined by $h_q(\boldsymbol{\theta})$ for a given node are first weighted and then summed. The latter procedure is done by the activation function K to generate the output for the particular node. In most instances, the sigmoidal function is used as the choice for K. A bias, which serves as a threshold for the argument of K, is generally introduced within the hidden and output layers and is accounted for by the difference between the predicted output by the ANN and the observed output data used to train the ANN. Further details on ANNs can be found in [80, 259].

The ANN used in this study was previously constructed by Dr. Roberto Rocchetta and was implemented in [1]. Its architecture comprises of 3 layers: 1 input-layer with 2 nodes, 1 hidden-layer with 10 nodes, and 1 output-layer with 6 nodes [1]. For the purpose of calibrating the ANN, 103 simulated values of the response frequencies $\{\omega_1, \ldots, \omega_6\}$ are obtained from 103 values of the mass positions $\{pm_1, pm_2\}$ in a series of computations involving a high-fidelity Finite element model of the aluminium frame structure which were performed by Dr. Peng Liang [4]. The resulting 103 sets of simulated data are presented in the form of scatterplots as shown in Figures 5.2a and 5.2b.

The calibration of the ANN was performed via the Feed-Forward Back-Propagation algorithm [260] with a sigmoidal activation function. Of the 103 sets of training data, 70 % was used to train the ANN, 15 % for validation, and 15 % for testing [1]. The calibration procedure took 0.513 s and yielded a regression coefficient R of: 0.9999 for training; 0.9993 for testing; and 0.9994 for validation. As the regression coefficients are all close to 1, these indicate that the existing ANN architecture is well calibrated. To illustrate this, the regression plots with respect to the training, testing, and validation data of the calibrated ANN are provided in Figure 5.3.

5.3 Bayesian model updating

In [1], 3 likelihood functions are presented and used to perform Bayesian model updating. Defining M_v being the model output for ω_v, details on the 3 likelihood functions are sum-



(a) Scatterplot of the 103 simulated values of $\{pm_1, pm_2\}$.

(b) Resulting scatterplot matrix of the response frequencies obtained from the FEM.

Figure 5.2: Simulated data obtained for the calibration of the ANN.

marised in Table 5.2. To address the uncertainty associated with the appropriate choice of the likelihood function L_m (for $m = 1, \ldots, 3$), an approach would be to perform a Bayesian model selection to determine the most probable likelihood function given the observed data [51]. However, there is also a need to introduce a degree of robustness in the estimates of

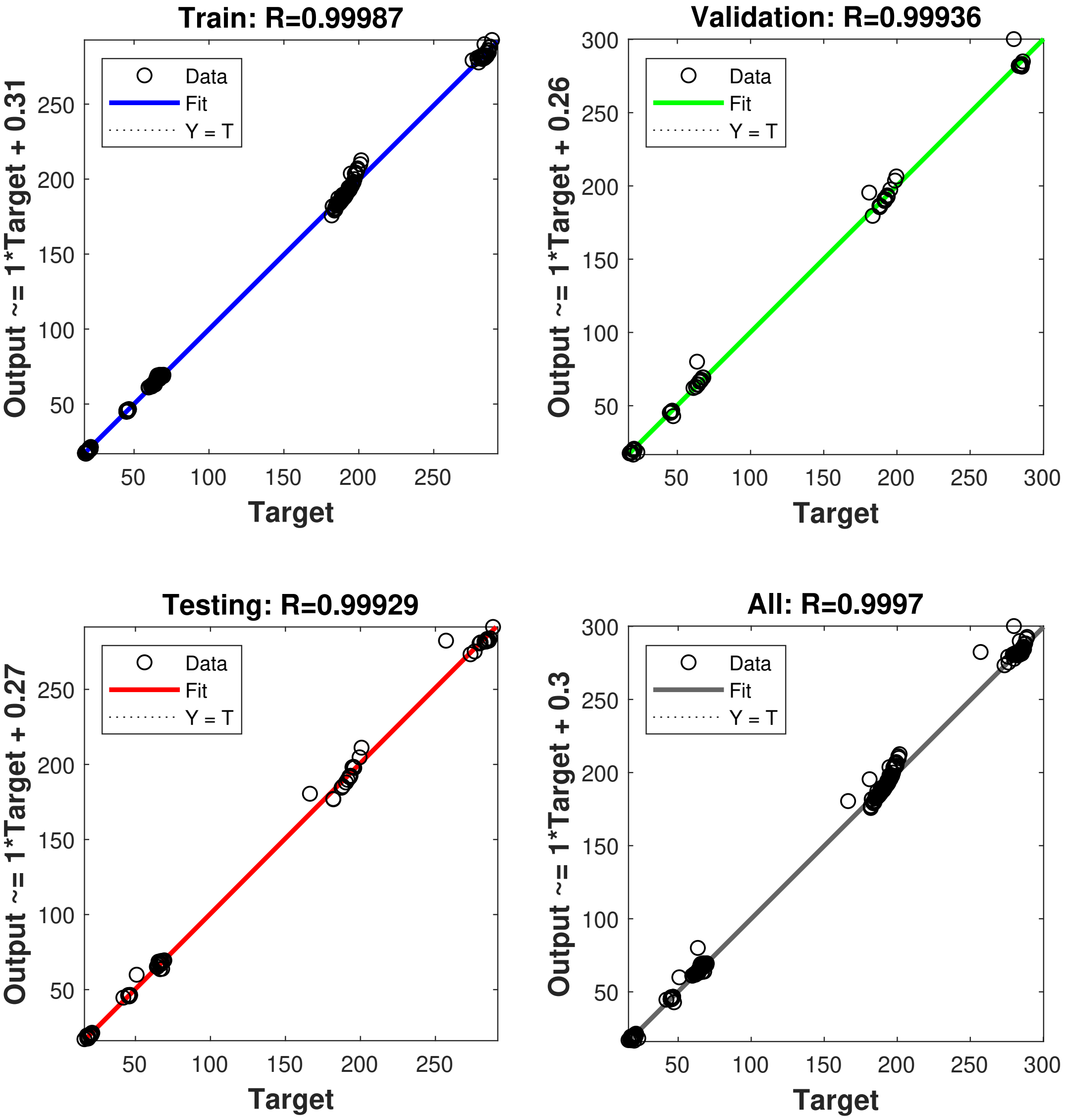


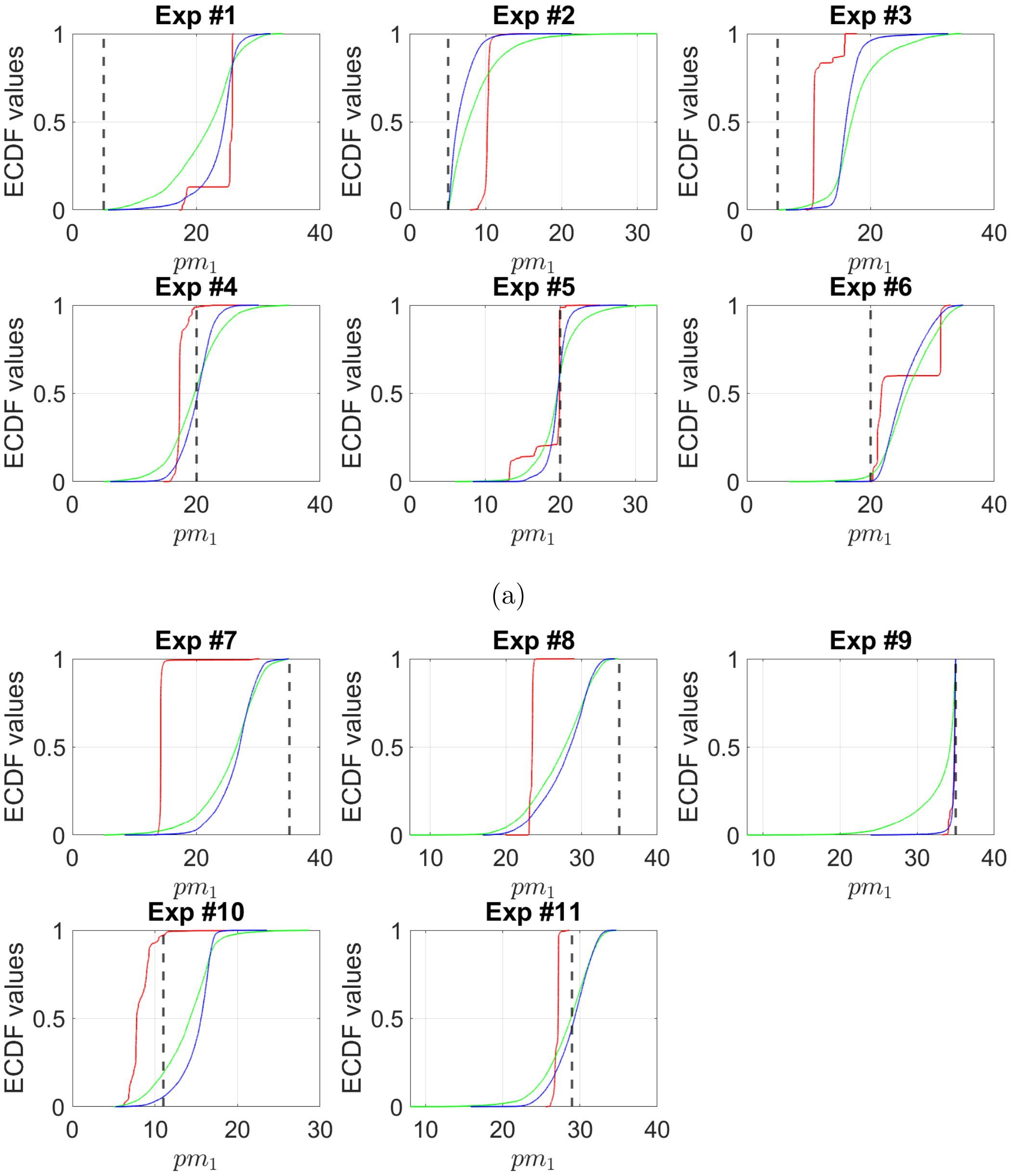
Figure 5.3: Regression plots of the calibrated ANN.

pm_1 and pm_2 by accounting for the incertitude over the choice of likelihood functions (i.e. the model uncertainty) rather than just the most probable likelihood function alone. Hence, the Robust Bayes (RB) framework is implemented to infer pm_1 and pm_2 [261, 262]. Using this framework, 3 distinct posteriors are derived by combining the prior with each of the likelihood function. Let the m^{th} posterior $P_m(\boldsymbol{\theta}|\boldsymbol{D}, M)$ be defined as the product of the prior and likelihood L_m. From there, a P-box [34, 36] can be constructed from the Empirical Cumulative Distribution Function (ECDF) of the posteriors where the 95 % Credible Interval (CI) for pm_1 and pm_2 are obtained. Such analysis is performed for all 11 sets of experiments from the data presented in Table 5.1.

Symbol	Type	$P(\boldsymbol{D}\|\boldsymbol{\theta}, M)$
L_1	Normal Distribution	$\prod_{v=1}^{6} \frac{1}{\sigma_v \cdot \sqrt{2\pi}} \cdot \exp\left[-\frac{(\omega_v - M_v)^2}{2 \cdot {\sigma_v}^2}\right]$
L_2	Inverse Squared Error	$\prod_{v=1}^{6} 1 - \exp\left[-\frac{1}{(\omega_v - M_v)^2}\right]$
L_3	Inverse Error	$\prod_{v=1}^{6} 1 - \exp\left[-\sqrt{\frac{1}{(\omega_v - M_v)^2}}\right]$

Table 5.2: The likelihood functions employed in [1] for the Bayesian inference of pm_1 and pm_2.

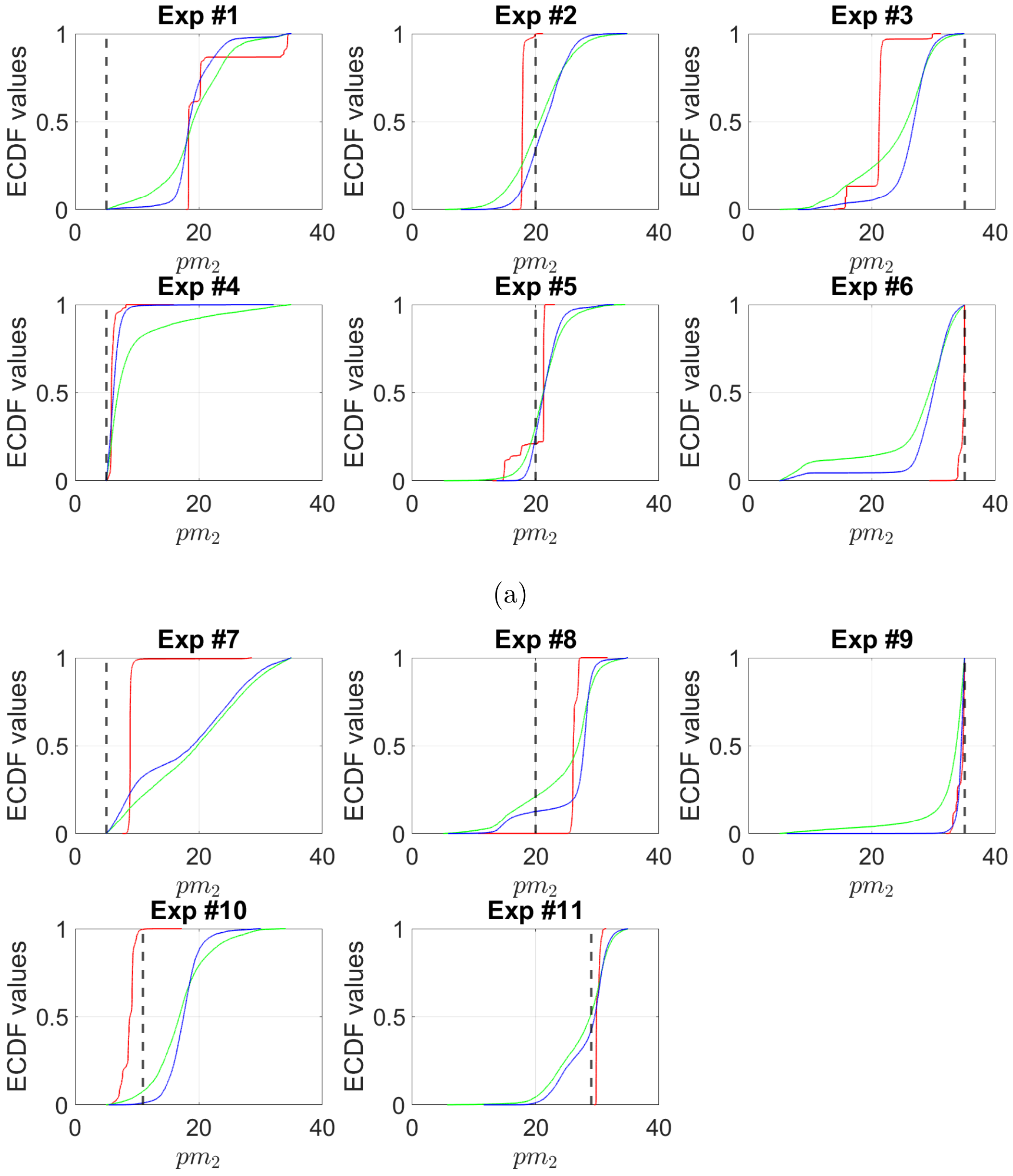
The priors for pm_1 and pm_2 are set as Uniform priors taking values between 5.0 cm and 35.0 cm. In addition, the measurement “noise” σ_v corresponding to the natural frequency ω_v, for $v = 1, \ldots, 6$, are also set as epistemic parameters to be inferred. The prior for each σ_v is also set as a Uniform prior taking values between 0.001 Hz and 100.0 Hz. Given that likelihood function L_1 is a function of σ_v, and that the likelihood functions L_2 and L_3 are independent of σ_v, this brings the total number of inferred parameters to 8 for the case of L_1 (i.e. $\boldsymbol{\theta} = (pm_1, pm_2, \sigma_1, \ldots, \sigma_6)$) and 2 for the case of L_2 and L_3 (i.e. $\boldsymbol{\theta} = (pm_1, pm_2)$) . For this problem, it will be assumed that the inferred parameters are independent from one another.



(a)

(b)

Figure 5.4: P-boxes for pm_1 for the respective experiments constructed from the ECDFs of $P_1(\boldsymbol{\theta}|\boldsymbol{D}, M)$ (red), $P_2(\boldsymbol{\theta}|\boldsymbol{D}, M)$ (green), and $P_3(\boldsymbol{\theta}|\boldsymbol{D}, M)$ (blue). The black dotted vertical line denotes the true value.



(a)

(b)

Figure 5.5: P-boxes for pm_2 for the respective experiments constructed from the ECDFs of $P_1(\boldsymbol{\theta}|\boldsymbol{D}, M)$ (red), $P_2(\boldsymbol{\theta}|\boldsymbol{D}, M)$ (green), and $P_3(\boldsymbol{\theta}|\boldsymbol{D}, M)$ (blue). The black dotted vertical line denotes the true value.

5.3.1 Results and discussions

From each $P_m(\boldsymbol{\theta}|\boldsymbol{D}, M)$, $N = 10000$ samples are generated using the TEMCMC sampler to ensure sufficient degree of accuracy when accounting for the tail quantile estimate. The

P-boxes obtained for each experiment are presented in Figure 5.4 for the case of pm_1, and Figure 5.5 for the case of pm_2. The numerical results of the 95 % CI obtained from the P-boxes for pm_1 and pm_2 for each experiment are summarised in Table 5.3. As shown in Table 5.3, the 95 % CI obtained for pm_1 generally encompasses the true value for most experiments with the exception of experiments where the true values of pm_1 are 5 cm and 35 cm. This is also observed for the case of the 95 % CI obtained for pm_2. A possible reason for such observation is as follows: Based on Figure 5.2a, it can be seen that most of the input training samples used to train the ANN are situated in the interval $[15, 25]$ cm for both pm_1 and pm_2. Beyond this sample region defined by the aforementioned interval, the samples are sparsely distributed. As such, in cases where the value(s) of either or both pm_1 and pm_2 fall(s) outside the interval of $[15, 25]$ cm, the resulting ANN is unable to mimic the general behaviour displayed by the FE model. This explanation can also be supported by Figure 5.3 where it can be seen that the training, testing, and validation samples are all not distributed evenly along the regression plot. In fact, for all the regression plots in Figure 5.3, the samples are clustered in distinct groups to which such observation is a result of most of the training input samples existing within the interval $[15, 25]$ cm for both pm_1 and pm_2. Hence, this results in the poor prediction performance by the ANN in the case where the true values of either or both pm_1 and pm_2 is/are 5 cm and 35 cm. Such behaviour by the ANN has also been highlighted in [1].

Exp. no.	True value pm_1 $[cm]$	95 % CI pm_1 $[cm]$	True value pm_2 $[cm]$	95 % CI pm_2 $[cm]$
1	5	$[9.26, 29.93]$	5	$[6.94, 34.37]$
2	5	$[5.08, 17.25]$	20	$[11.80, 29.71]$
3	5	$[9.87, 28.82]$	35	$[10.69, 31.95]$
4	20	$[11.06, 28.50]$	5	$[5.09, 29.49]$
5	20	$[13.22, 26.19]$	20	$[14.83, 28.70]$
6	20	$[19.14, 33.53]$	35	$[6.32, 35.00]$
7	35	$[13.90, 32.40]$	5	$[5.63, 33.56]$
8	35	$[18.92, 32.87]$	20	$[11.47, 32.15]$
9	35	$[24.26, 34.99]$	35	$[13.32, 34.99]$
10	11	$[6.39, 19.57]$	11	$[6.58, 27.86]$
11	29	$[21.01, 33.00]$	29	$[18.94, 33.47]$

Table 5.3: Results of the 95 % CI obtained from the P-boxes for pm_1 and pm_2 for each experiment.

The resulting sampling time elapsed, number of iterations, and range of acceptance rates across iterations are summarised in Tables 5.4 to 5.6. As seen from the table, the acceptance rate interval by the TEMCMC sampler generally encompasses the optimal range of $[0.15, 0.50]$ for all experiments and choice of combined likelihood functions. This highlights the robustness of the proposed adaptive tuning algorithm in moderating the acceptance rates of the sampler. In addition, it can also be seen that the number of iterations required by the TEMCMC sampler to generate samples from $P_1(\boldsymbol{\theta}|\boldsymbol{D}, M)$ is the highest while a similar number of iterations was required for $P_2(\boldsymbol{\theta}|\boldsymbol{D}, M)$ and $P_3(\boldsymbol{\theta}|\boldsymbol{D}, M)$. A reason to account for the significantly higher number of iterations involved in sampling from $P_1(\boldsymbol{\theta}|\boldsymbol{D}, M)$ is due to the relative rapid posterior contraction, as suggested by its ECDF plots in Figures 5.4 and 5.5 which exhibit a steeper gradient in general compared to those of $P_1(\boldsymbol{\theta}|\boldsymbol{D}, M)$ and $P_3(\boldsymbol{\theta}|\boldsymbol{D}, M)$.

Exp. no.	$P_1(\boldsymbol{\theta}\|\boldsymbol{D}, M)$		
	Time $[s]$	Iterations	Acceptance
1	3.83×10^3	12	$[0.316, 0.479]$
2	4.13×10^3	12	$[0.353, 0.519]$
3	3.27×10^3	10	$[0.332, 0.475]$
4	3.91×10^3	12	$[0.335, 0.478]$
5	4.17×10^3	13	$[0.329, 0.488]$
6	3.78×10^3	12	$[0.336, 0.481]$
7	3.10×10^3	9	$[0.354, 0.517]$
8	4.03×10^3	12	$[0.312, 0.502]$
9	3.65×10^3	11	$[0.338, 0.483]$
10	4.00×10^3	13	$[0.305, 0.478]$
11	3.87×10^3	12	$[0.318, 0.479]$

Table 5.4: Summary of the sampling time elapsed, number of iterations, and range of acceptance rates across iterations by the TEMCMC sampler for $P_1(\boldsymbol{\theta}|\boldsymbol{D}, M)$.

5.4 Chapter conclusion

The chapter has presented an experimental example involving a $2DoF$ Shear aluminium frame structure with 2 moveable masses to simulate the damage locations within a given structure. Through this example, the performance of the proposed Transitional Ensemble Markov Chain Monte Carlo sampler is validated in identifying the mass positions within a $2DoF$ Shear aluminium frame structure given a set of response frequencies obtained via a hammer impact test in a previous experimental campaign. In doing so, this simulates the

Exp. no.	$P_2(\boldsymbol{\theta}\|\boldsymbol{D}, M)$		
	Time $[s]$	Iterations	Acceptance
1	800.23	2	$[0.444, 0.493]$
2	8.48×10^2	3	$[0.434, 0.485]$
3	1.26×10^3	2	$[0.447, 0.495]$
4	8.53×10^2	3	$[0.434, 0.486]$
5	1.23×10^3	3	$[0.440, 0.494]$
6	1.27×10^3	2	$[0.437, 0.490]$
7	8.36×10^2	2	$[0.444, 0.493]$
8	8.76×10^2	2	$[0.446, 0.488]$
9	1.54×10^3	4	$[0.409, 0.485]$
10	1.25×10^3	3	$[0.416, 0.508]$
11	1.22×10^3	3	$[0.461, 0.505]$

Table 5.5: Summary of the sampling time elapsed, number of iterations, and range of acceptance rates across iterations by the TEMCMC sampler for $P_2(\boldsymbol{\theta}|\boldsymbol{D}, M)$.

Exp. no.	$P_3(\boldsymbol{\theta}\|\boldsymbol{D}, M)$		
	Time $[s]$	Iterations	Acceptance
1	1.28×10^3	2	$[0.448, 0.481]$
2	1.65×10^3	3	$[0.424, 0.502]$
3	1.29×10^3	2	$[0.434, 0.499]$
4	1.64×10^3	3	$[0.412, 0.481]$
5	1.68×10^3	3	$[0.433, 0.500]$
6	1.26×10^3	2	$[0.457, 0.498]$
7	8.68×10^2	2	$[0.441, 0.490]$
8	1.31×10^3	2	$[0.455, 0.497]$
9	1.92×10^3	4	$[0.417, 0.484]$
10	1.28×10^3	3	$[0.423, 0.491]$
11	1.26×10^3	3	$[0.441, 0.502]$

Table 5.6: Summary of the sampling time elapsed, number of iterations, and range of acceptance rates across iterations by the TEMCMC sampler for $P_3(\boldsymbol{\theta}|\boldsymbol{D}, M)$.

procedure of damage detection in structural health monitoring.

To identify the mass positions of the 2 moveable masses of the frame structure using Bayesian inference, 3 different likelihood functions are considered. To loosen the assumption on the form of the likelihood function, and account for the uncertainty over the choice likelihood functions to be used, the Robust Bayes approach has been implemented. Through such approach the Transitional Ensemble Markov Chain Monte Carlo sampler is able to generate posterior estimates whose 95 % Credible Intervals generally envelops the true mass positions for most experiments.

In addition, a discussion is also provided regarding the sampling performance of the Transitional Ensemble Markov Chain Monte Carlo sampler. There are 2 aspects which are highlighted: 1) the acceptance rates of the sampler generally fall within the bounds of $[0.15, 0.50]$ (i.e. see Section 4.3.3 of the dissertation) which highlight the robustness of the proposed adaptive tuning algorithm within the sampler in moderating such quantity; and 2) the number of sampling iterations is the highest in sampling from the posterior constructed using the Normal distribution likelihood function compared to using the Inverse Squared Error and Inverse Error likelihood functions.

Through the results presented in the chapter, it highlights the applicability of the proposed Transitional Ensemble Markov Chain Monte Carlo sampler in performing off-line structural health monitoring in the context of the experimental example that is used to simulate a structure subjected to damage. This, however, is subjected to data samples used to train the surrogate model (i.e. the ANN) used for the Bayesian model updating procedure, and its subsequent training, testing and validation performances. A discussion to which is provided in Section 5.3.1 of the chapter.

Finally, the MATLAB codes for the example presented in the chapter has been made accessible on GitHub: `https://github.com/Adolphus8/Transitional_Ensemble_MCMC.git`

Chapter 6

Distribution-free Analysis of a Dynamical Black-box system under Uncertainty

Abstract

The chapter presents another real-world application of the Transitional Ensemble Markov Chain Monte Carlo sampler, presented in Chapter 4, in a problem that is based on the 2019 edition of the NASA-Langley Uncertainty Quantification challenge. The challenge presented in this problem involves the analysis on a set of critical systems while lacking available data. This results in a large degree of uncertainty and the need for uncertainty quantification tools so as to make risk-informed decisions. As such, in the chapter, a framework is proposed that is aimed that achieving 2 objectives: 1) to calibrate an Uncertainty Model consisting of both aleatory and epistemic parameters; and 2) to reduce the uncertainty of the calibrated Uncertainty Model.

6.1 Background

6.1.1 Research context

The design of critical safety systems is often associated with the availability of limited data. Despite such challenge posed, the system needs to be designed in order to cope with the unavoidable uncertainty. Such uncertainty can be classified as either aleatory or epistemic uncertainty [24, 25] to which their respective definition are provided in Section 1.1 of the dissertation. It needs to be noted that the aleatory and epistemic uncertainty can refer to the same physical quantity and, therefore, such classification can become fuzzy. In fact, the aleatory uncertainty can be seen as the remaining uncertainty after a campaign, aimed at reducing the epistemic uncertainty, is performed.

The design of systems under uncertainty requires the availability of robust and efficient tools for uncertainty characterisation and quantification. In order to check the availability of discipline independent tools and applicability of such tools, NASA-Langley proposed a new Uncertainty Quantification Challenge problem in 2019 [26] with the purpose of modelling the dynamic behaviour of a system, analysing its operational reliability, and devising an improved design configuration for the system under uncertainty. This Uncertainty Quantification Challenge problem follows from the success of the previous edition in 2013 [263].

In this challenge, a "Black-box" computational model of a physical system is used to evaluate and improve its reliability. Unlike the previous challenge [263], the Uncertainty Model (UM) to the respective aleatory input parameters are completely unknown and they are to be derived by the participants. In addition, the response of the system is time-dependent providing a realistic setting under which different tasks will be addressed. A description to the physical system of interest and the research problem is provided in Section 6.1.2.

6.1.2 Problem statement

The physical system of interest is characterised by a design point $\boldsymbol{\Theta}$ with 9 real components (i.e. $\boldsymbol{\Theta} \in \mathbb{R}^9$). The Black-box computational model of the physical system has a set of uncertain model parameters comprising of elements $\boldsymbol{a}$ and $\boldsymbol{e}$ representing the vector of aleatory and epistemic parameters respectively [26]. For this problem, $\boldsymbol{a}$ has 5 real components while $\boldsymbol{e}$ has 4 real components. This set of uncertain model parameters is denoted by δ which is represented by a UM.

The aleatory space A is represented as $\boldsymbol{a} \sim \pi_{\boldsymbol{a}}$ whereby $\pi_{\boldsymbol{a}}$ is the joint density function. The initial aleatory space is $A_0 = [0, 2]^5$. The epistemic space E is represented as $\boldsymbol{e} \sim E$. The initial epistemic space is $E_0 = [0, 2]^4$. Hence, the UM for δ is fully characterized by: $\langle \pi_{\boldsymbol{a}}, E \rangle$.

The system consists of a set of interconnected subsystems for which δ is concentrated in one of these subsystems. This subsystem is modelled by a Black-box model function $\hat{y} = y_{fun}(\boldsymbol{a}, \boldsymbol{e}, t)$, where $t \in [0, 5]$ s is the time parameter. The output of this subsystem is represented as a discrete time history: $y^l(t) = [y^l(0), y^l(dt), \dots, y^l(5000 \cdot dt)]$, where $l = 1, \dots, 100$, and $dt = 0.001$ s. This yields a total of $N_t = 5001$ data of $y^l(t)$ per given l and the entire time history data is denoted as $D_1 = \{y^l(t)\}_{l=1,\dots,100}$. As an illustration, the graphical plot of the time history data D_1 is provided in Figure 6.1.

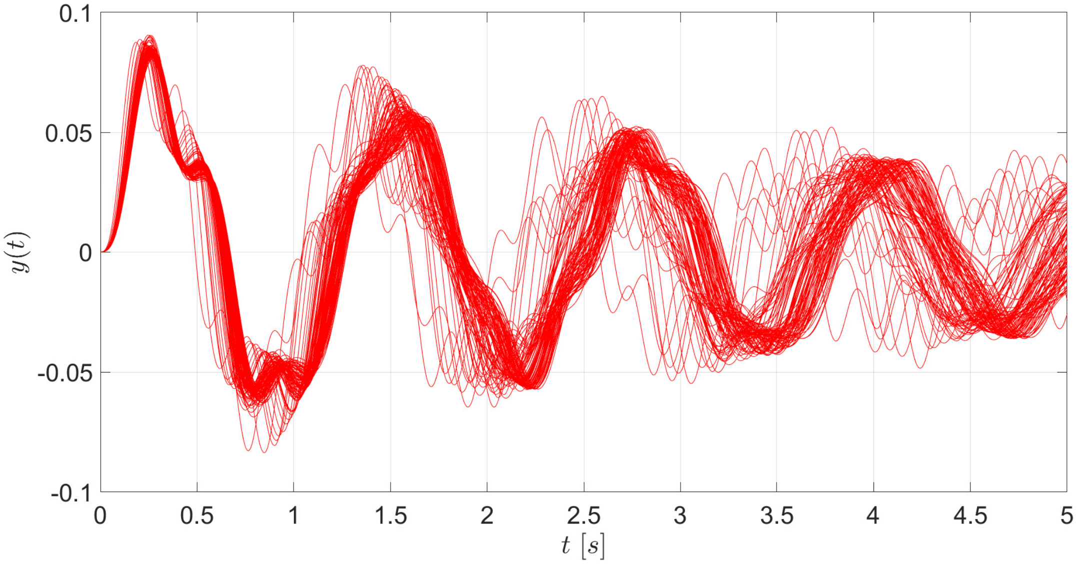


Figure 6.1: Graphical illustration of the time history data D_1.

The tasks involved this challenge are summarised as follows [26]:

1) To create an UM for δ;

2) To decide a limited number of refinements (up to 4) on the epistemic parameters;

3) To perform a reliability analysis on a given design point $\boldsymbol{\Theta}$;

4) To identify a new $\boldsymbol{\Theta}$ with improved reliability;

5) To improve the UM for δ and $\boldsymbol{\Theta}$ given observations of the integrated system.

Full details on the challenge can be found in [26].

In the interest of keeping in-line with the context of the dissertation defined in Chapter 1, only Tasks 1 and 2 of this challenge are presented in the chapter. Results and discussions to the rest of the challenge tasks can be found in [264].

6.2 Task 1: Calibration of the Uncertainty model

The UM is calibrated via the Bayesian model updating technique using the available data D_1. This provides a probabilistic approach through which the joint distribution function $\pi_{\boldsymbol{a}}$ can be identified. However, the Bayesian model updating approaches are inapplicable in reducing the epistemic uncertainty when such uncertainty is represented by intervals or any other non-probabilistic form as per the case in this challenge. To overcome this limitation, a computational trick would be to model the interval, representing the epistemic uncertainty, following a distribution which is usually assumed to be Uniform for simplicity. This is so as to be able to generate samples from such intervals and find the bounds to the uncertain response signal. It needs to be noted that the form of the distribution does not affect the results since only the bounds of the response is of interest and there is no probability associated to the output. Such approach has been proposed and implemented in [265]. Therefore, the epistemic parameters e_{i_e}, for $i_e = 1, \ldots, 4$, are assumed to be independent between one another and their respective priors modelled by a Uniform distribution with bounds defined by the epistemic space E_0 (see Section 6.1.2).

Within the Bayesian model updating framework, the likelihood function $P(\boldsymbol{D}|\boldsymbol{\theta}, M)$ is an important component since it reflects the degree of error between the set of observed data D_1 and the "Black-box" model $\hat{y}$. The latter is a function of the input model parameters which are to be inferred (i.e. see Section 2.2.2). In the case of D_1, the data set is multi-dimensional which comprises of $N_t \times 100$ entries. Assuming independence between each of the observed data entry, $P(\boldsymbol{D}|\boldsymbol{\theta}, M)$ is defined as per in Eq. (2.4). However, such assumption leads to the loss of information on the correlation of the data D_1 between successive t. In addition, the use of density estimators, such as the Kernel Density Estimation approach, requires a large number of model evaluations to perform a sufficiently accurate estimation on the distribution on the data. This can be a computationally expensive procedure. To reduce the computational cost, the Approximate Bayesian Computation (ABC) approach [110, 266] is implemented which involves the use of an approximate likelihood function [267]. Such approximate likelihood function employs the use of a stochastic distance metric d which quantifies the difference between the distribution of the observed data and the model output. The principle behind such approach is that the approximate likelihood function would return a high value when d is small and returns a low value when d is large, thereby penalising the realizations of $\boldsymbol{\theta}$ that lead to the latter. The choice of $P(\boldsymbol{D}|\boldsymbol{\theta}, M)$ used in this problem is the approximate Gaussian function defined as [267]:

$$P(\boldsymbol{D}|\boldsymbol{\theta}, M) \propto \exp\left(-\frac{d^2}{\varepsilon^2}\right) \tag{6.1}$$

where ε is the width factor of the approximate Gaussian function.

From the resulting posterior, advanced sampling techniques are employed to obtain samples (i.e. see Chapters 2 and 3). In this work, the TEMCMC sampler is chosen and implemented for 3 key reasons: 1) the algorithm is able to sample from complex-shaped posteriors via "transitional" distributions P^j (i.e. see Eq. (3.6)); 2) it can sample from high-dimensional posteriors (i.e. up to 24 dimensions) [227]; and 3) it computes the evidence $P(\boldsymbol{D}|M)$ which makes the algorithm useful in model selection problems [51].

For this task, different strategies are adopted to represent the aleatory uncertainty and metrics are used to compare between distributions. As a result, 2 approaches aimed at reducing the uncertainty of the UM are developed: 1) a conservative (i.e. low-risk) approach; and 2) a more aggressive (i.e. high-risk) approach. Details on the respective approaches will be discussed in Sections 6.2.2 and 6.2.3 of the chapter.

6.2.1 Proposed approaches

To determine the UM for $\boldsymbol{a}$, 5 possible distribution types for $\pi_{\boldsymbol{a}}$ are identified and listed in Table 6.1. It needs to be highlighted that the choice of the Staircase Density Function (SDF) presents a distribution-free approach contrary to the other choice of distributions presented in the table. A key strength of SDFs lies in its flexibility in describing a wide range of density shapes, including highly-skewed and/or multi-modal distributions. This makes them highly applicable in modelling the marginal distributions of the aleatory parameters whose density shapes are unknown a priori. In this analysis it is assumed that: 1) the marginal distribution of all the aleatory uncertainties belong to the same distribution class; 2) no dependency exists between all a_{i_a} (i.e. no correlation matrix used).

Aleatory model	Distribution type	Prior distribution parameters
$\pi_{\boldsymbol{a}}^1$	Beta($\alpha_{i_a}, \beta_{i_a}$)	α_{i_a} (Shape parameter 1): $U[0, 100]$ β_{i_a} (Shape parameter 2): $U[0, 100]$
$\pi_{\boldsymbol{a}}^2$	Truncated Normal(μ_{i_a},σ_{i_a}) $[TN(a_{i_a}; \mu_{i_a}, \sigma_{i_a})]$	μ_{i_a} (Mean of a_{i_a}): $U[0, 2]$ σ_{i_a} (Standard deviation of a_{i_a}): $U[0.01, 2]$
$\pi_{\boldsymbol{a}}^3$	Truncated Lognormal $[TLN(a_{i_a}; \mu_{i_a}, \sigma_{i_a})]$	μ_{i_a} (Mean of $log(a_{i_a})$): $U[-10, 10]$ σ_{i_a} (Standard deviation of $log(a_{i_a})$): $U[0.01, 5]$
$\pi_{\boldsymbol{a}}^4$	Truncated Gamma $[TG(a_{i_a}; \alpha_{i_a}, \beta_{i_a})]$	α_{i_a} (Shape parameter): $U[0, 10]$ β_{i_a} (Scale parameter): $U[0, 10]$
$\pi_{\boldsymbol{a}}^5$	Staircase Density Function $[SDF(a_{i_a}; \mu_{i_a}, (m_2)_{i_a}, (m_3)_{i_a}, (m_4)_{i_a})]$	μ_{i_a} (Distribution mean of a_{i_a}): $U[0, 2]$ $(m_2)_{i_a}$ (2^{nd} central moment of a_{i_a}): $U[0, 1]$ $(m_3)_{i_a}$ (3^{rd} central moment of a_{i_a}): $U[-\frac{4}{3\sqrt{3}}, \frac{4}{3\sqrt{3}}]$ $(m_4)_{i_a}$ (4^{th} central moment of a_{i_a}): $U[0, \frac{4}{3}]$

Table 6.1: Distribution type with the Uniform prior bounds of its corresponding parameters for each aleatory model $\pi_{\boldsymbol{a}}$.

6.2.2 Distribution-based approach

For the case of $\pi_{\boldsymbol{a}}^1$ to $\pi_{\boldsymbol{a}}^4$, the distribution parameters add an additional 10 inferred parameters, while $\pi_{\boldsymbol{a}}^5$ adds an additional 20 inferred parameters. Each of these parameters are assigned a Uniform prior with bounds, stated in Table 6.1, chosen to ensure sufficient degrees of freedom in the model calibration. It is also assumed that these parameters are independent from one another. This brings the total number of inferred parameters to 24 for the case of the SDF, and 14 for the rest of the distributions.

For the case of $\pi_{\boldsymbol{a}}^1$ to $\pi_{\boldsymbol{a}}^4$, there is a need to reduce the size of the data to reduce the computation cost in evaluating $P(\boldsymbol{D}|\boldsymbol{\theta}, M)$. To achieve this, the Fast Fourier Transformation (FFT) procedure is performed on D_1 for each l according to [227, 268, 269]:

$$y^l(t) = \sum_{q=0}^{5000} C_q^l \cdot \exp\left[-i \cdot q \cdot \omega_0 \cdot t\right] \tag{6.2}$$

where $\omega_0 = \frac{2 \cdot \pi}{5001}$, and C_q^l is the numerical coefficient with real and imaginary components denoted as $\mathrm{Re}(C_q^l)$ and $\mathrm{Im}(C_q^l)$ respectively. From which, the amplitude A_q^l and phase angles ϕ_q^l are obtained as follows:

$$A_q^l = \sqrt{\mathrm{Re}(C_q^l)^2 + \mathrm{Im}(C_q^l)^2} \tag{6.3}$$

$$\phi_q^l = \text{atan2}\left[\frac{\text{Im}(C_q^l)}{\text{Re}(C_q^l)}\right] \tag{6.4}$$

To remove the periodicity associated with the values of ϕ_q^l, a phase shift is introduced such that a factor of 2π rad is added or subtracted whenever the jump between consecutive phase angles is greater than π rad. This is achieved using the *unwrap* function in MATLAB to ensure the jump between any consecutive phase angles is always less than π rad. In doing so, it ensures the monotonic behaviour of ϕ_q^l and simplifies its subsequent computation for $P(\boldsymbol{D}|\boldsymbol{\theta}, M)$. When this is done, the frequency spectra of A_q^l and ϕ_q^l are obtained as shown in Figure 6.2 where it can be observed that beyond frequencies $\omega > 5.80\ Hz$, the values of A_q^l do not show any additional perturbations for all l, thereby allowing those data to be discarded. Hence, only 30 values of ω between 0 Hz and 5.80 Hz are considered for both A_q^l and ϕ_q^l. Let this set of values of ω be denoted as ω_n, for $n = 1, \ldots, 30$. This effectively reduces the total number of data for model calibration from 500100 to 6000 (i.e. 3000 for A_q^l and 3000 for ϕ_q^l).

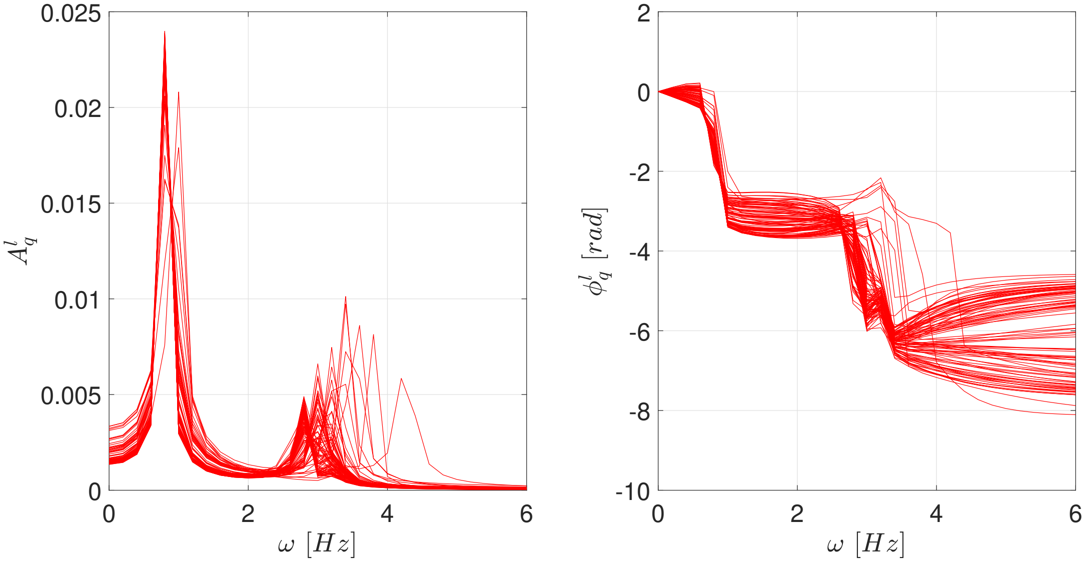


Figure 6.2: Illustration of the frequency spectra obtained from D_1 via FFT.

To account for the variability of A_q^l and ϕ_q^l at each ω_n, the stochastic distance metric d for $P(\boldsymbol{D}|\boldsymbol{\theta}, M)$ is the Wasserstein distance defined as [270]:

$$d_W = \int_{-\infty}^{\infty} |F_D(x) - F_{\hat{y}}(x)| \cdot dx \tag{6.5}$$

whereby $F_D(x)$ and $F_{\hat{y}}(x)$ are the respective ECDFs of the data (i.e. A^l_q and ϕ^l_q) and the stochastic model output of $\hat{y}$ at a given ω_n, while x is the variable denoting either A^l_q or ϕ^l_q. In essence, d_W quantifies the enclosed area between both ECDFs. The smaller d_W is, the higher the degree of similarity between the ECDFs of the data and the stochastic prediction by $\hat{y}$ [271]. Using Eq. (6.1), $P(\boldsymbol{D}|\boldsymbol{\theta}, M)$ is defined as:

$$P(\boldsymbol{D}|\boldsymbol{\theta}, M) = \prod_{n=1}^{30} \exp\left[-\left(\frac{d^A_{W,n}}{\varepsilon^A_n}\right)^2 - \left(\frac{d^\phi_{W,n}}{\varepsilon^\phi_n}\right)^2\right] \tag{6.6}$$

whereby the values of ε^A_n and ε^ϕ_n are approximated by the standard deviations of A^l_q and ϕ^l_q respectively at ω_n. Independence is assumed between data sets to reduce computational costs in computing $P(\boldsymbol{D}|\boldsymbol{\theta}, M)$. However, it needs to be highlighted that in reality, there exists dependencies between the identified ω_n for each l^{th} sequence. To compute $P(\boldsymbol{D}|\boldsymbol{\theta}, M)$, 100 model evaluations by $\hat{y}$, per given set of model inputs $\{\boldsymbol{a}, \boldsymbol{e}\}$, are needed to construct $F_{\hat{y}}(x)$.

6.2.3 Distribution-free approach

For the case of $\pi^5_{\boldsymbol{a}}$, the distribution is defined by the SDF as [272]:

$$\pi_{\boldsymbol{a}} = \begin{cases} h_{i_b} \ \forall \, \boldsymbol{a} \in ((i_b - 1)\cdot\kappa, i_b\cdot\kappa]^5 & \text{, for } 1 \leq i_b \leq N_b \\ 0 & \text{, otherwise} \end{cases} \tag{6.7}$$

where $N_b = 50$ is the number of bins, h_{i_b} is the height of the SDF in the $i_b{}^{th}$ bin, and $\kappa = \frac{2}{N_b}$ is the length of each sub-interval. It needs to be noted that $h_{i_b} \geq 0$ for all N_b bins and that their values are obtained by solving the following convex optimisation problem:

$$\hat{h}_{i_b} = \underset{h_{i_b} \geq 0}{\text{argmin}} \left\{ J(h) : \sum_{i_b=1}^{N_b} \int_{(i_b-1)\cdot\kappa}^{i_b\cdot\kappa} z \cdot h_{i_b} \cdot dz = \mu_{i_a}, \sum_{i_b=1}^{N_b} \int_{(i_b-1)\cdot\kappa}^{i_b\cdot\kappa} (z - \mu_{i_a})^r \cdot h_{i_b} \cdot dz = (m_r)_{i_a} \right\} \tag{6.8}$$

where $r = 0, 2, 3, 4$, and $J(h)$ is the cost-function. Details on $J(h)$ and Eq. (6.8) can be found in [272, 33].

To avoid a potential error in the implementation of the likelihood function for the Distribution-based approach being brought forward, a different set-up for $P(\boldsymbol{D}|\boldsymbol{\theta}, M)$ is used in this analysis, where a different stochastic distance is employed and the data is analyzed in the time domain. Unlike in the Distribution-based approach, the discrete Bhattacharyya distance [267] is employed as the stochastic distance metric d:

$$d_B = -\log \left\{ \sum_{i_{N_t}=1}^{n_b} \cdots \sum_{i_1=1}^{n_b} \sqrt{p_D(b_{i_1,\ldots,i_{N_t}}) \cdot p_{\hat{y}}(b_{i_1,\ldots,i_{N_t}})} \right\} \tag{6.9}$$

where $p_D(b_{i_1,\ldots,i_{N_t}})$ and $p_{\hat{y}}(b_{i_1,\ldots,i_{N_t}})$ are the Probability Mass Function (PMF) values of the data from D_1 and the stochastic model output from $\hat{y}$ respectively within the bin $b_{i_1,\ldots,i_{N_t}}$, and $n_b = 20$ is the number of bins used to compute the Bhattacharyya distance.

It needs to be highlighted that each bin has N_t coordinates as it is generated within a N_t-dimensional joint PMF space. Because of this, the resulting joint PMF space has an excessive number of dimensions for a direct evaluation of $P(\boldsymbol{D}|\boldsymbol{\theta}, M)$. This brings the need for a dimension-reduction procedure which is employed through the following steps [273]:

1. Define the window length $L_w = 50$ and divide the data set $\{y^l(t)\}_{l=1,\ldots,100}$ into $\left\lceil \frac{N_t}{L_w} \right\rceil$ distinct intervals where $\lceil \bullet \rceil$ is the ceil operator;

2. Compute the Root Mean Squared (RMS) values of each interval $\boldsymbol{R} = \left[R_1, \ldots, R_{\lceil \frac{N_t}{L_w} \rceil}\right]$ and generate the sample set of the RMS values $\boldsymbol{R}_D \in \mathbb{R}^{100 \times \lceil \frac{N_t}{L_w} \rceil}$ where:

$$\boldsymbol{R}_D = \left[\boldsymbol{R}_D^1, \ldots, \boldsymbol{R}_D^{\lceil \frac{N_t}{L_w} \rceil}\right], \text{ with } \boldsymbol{R}_D^{\nu} = \left[R_{1,\nu}, \ldots, R_{100,\nu}\right]^T$$

   for $\nu = 1, \ldots, \left\lceil \frac{N_t}{L_w} \right\rceil$ while $\boldsymbol{R}_{\hat{y}} \in \mathbb{R}^{N_{sim} \times \lceil \frac{N_t}{L_w} \rceil}$ where $N_{sim} = 1000$ the number of model evaluations by $\hat{y}$ per given set of model inputs $\{\boldsymbol{a}, \boldsymbol{e}\}$. It needs to be highlighted that the matrix structure of $\boldsymbol{R}_{\hat{y}}$ is similar to that of $\boldsymbol{R}_D$ with the exception of the number of row elements;

3. Evaluate d_B between sample sets $\boldsymbol{R}_D^{\nu}$ and $\boldsymbol{R}_{\hat{y}}^{\nu}$ for all ν;

4. Obtain the corresponding RMS values R_{d_B} and use it as the distance metric.

Consequently, $P(\boldsymbol{D}|\boldsymbol{\theta}, M)$ is re-expressed as:

$$P(\boldsymbol{D}|\boldsymbol{\theta}, M) = \exp\left(-\frac{{R_{d_B}}^2}{{\varepsilon_B}^2}\right) \tag{6.10}$$

where $\varepsilon_B = 0.01$.

6.2.4 Results and discussions

For this challenge problem, the model evaluation procedure by the given Black-box model $\hat{y}$ involves significant computational costs. In addition, there is also a need to ensure that the sample size is sufficiently large to obtain Monte Carlo estimates on the inferred parameters $\boldsymbol{\theta}$ within an acceptable degree of uncertainty. To account for these 2 factors, $N = 500$ samples are obtained from the resulting $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$.

Based on the analysis done for all aleatory models $\pi_{\boldsymbol{a}}$, 2 models are chosen on the basis of their quality of the results and for the subsequent purpose of comparison: $\pi_{\boldsymbol{a}}^1$ and $\pi_{\boldsymbol{a}}^5$. It needs to be highlighted, that $\pi_{\boldsymbol{a}}^1$ is chosen given its relatively higher value of the evidence $P(\boldsymbol{D}|M)$ compared to the other aleatory models used in the Distribution-based approach as shown in Table 6.2.

Aleatory model	$\pi_{\boldsymbol{a}}^1$	$\pi_{\boldsymbol{a}}^2$	$\pi_{\boldsymbol{a}}^3$	$\pi_{\boldsymbol{a}}^4$
$P(\boldsymbol{D}|M)$	3.2229×10^{-7}	2.1952×10^{-7}	5.0815×10^{-10}	1.0180×10^{-9}

Table 6.2: Results of the evidence computed via TEMCMC for each choice of model for $\pi_{\boldsymbol{a}}$.

To create the UM based on the information from the Bayesian model updating results, the following procedure is undertaken: For the aleatory space, the histograms of the distribution parameters of the given $\pi_{\boldsymbol{a}}$ are obtained from $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$. These histograms are converted into probability distribution functions using Kernel density estimation with a Gaussian kernel [274] and are normalised such that the distribution peak equals to 1. An illustration is provided using the distribution parameters for $\pi_{\boldsymbol{a}}^1$ as an example in Figure 6.3. From these results, the posterior distributions are interpreted as Fuzzy sets where different levels of statistical significance $L_c \in [0, 1]$ would yield intervals of varying width. Such approach has been proposed by Dubois *et. al* (2004) [275] and previously implemented in [36, 265] Here, intervals at $L_c = 0.50$ level of confidence are considered for both the $\pi_{\boldsymbol{a}}^1$ and $\pi_{\boldsymbol{a}}^5$ distribution parameters. The resulting intervals obtained would serve as shape parameter inputs of the respective aleatory model $\pi_{\boldsymbol{a}}$. This yields the P-boxes [34, 182] of $\pi_{\boldsymbol{a}}^1$ and $\pi_{\boldsymbol{a}}^5$ which are illustrated in Figure 6.4.

To define the epistemic space, the same procedure is done on the resulting histograms of e_1 to e_4 obtained through $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$ given the respective $\pi_{\boldsymbol{a}}$. These histograms are illustrated in Figure 6.5 for the respective set-up. However, $L_c = 0.05$ level of confidence is considered in the case of $\pi_{\boldsymbol{a}}^1$ while $L_c = 0.025$ level of confidence is considered in the case of

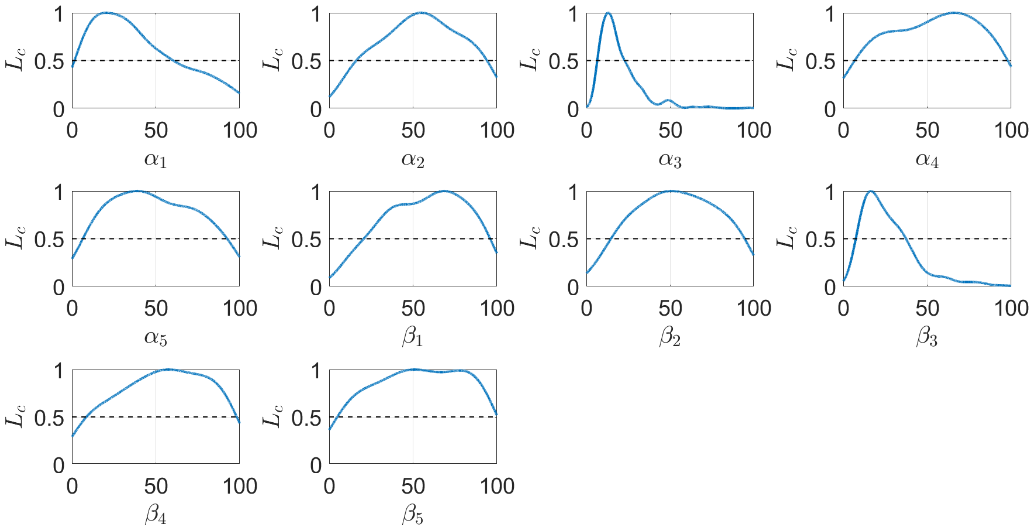


Figure 6.3: Illustration of the resulting distribution functions to the respective shape parameters of $\pi_{\boldsymbol{a}}^1$ (i.e. the joint Beta distribution) obtained via Kernel density estimates.

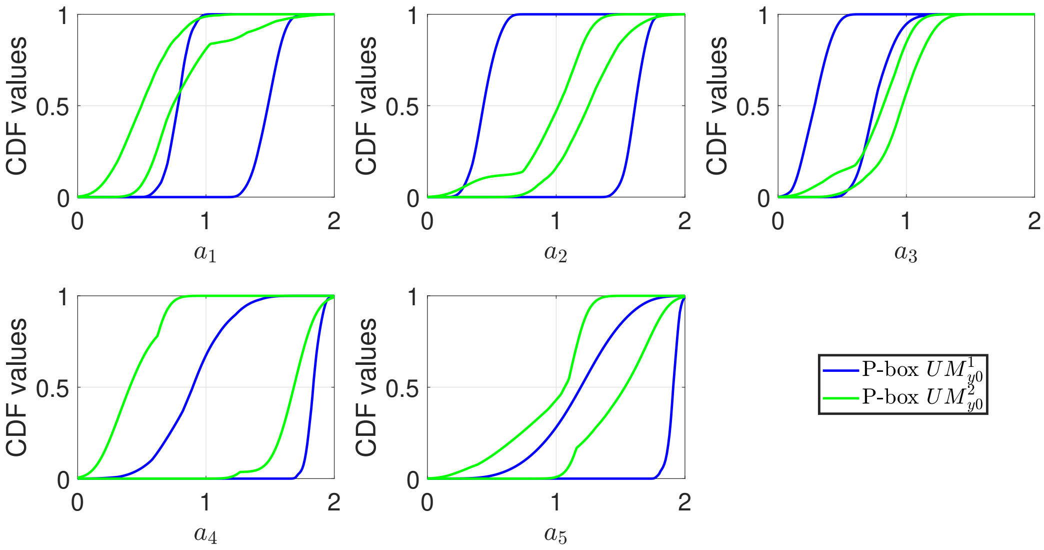


Figure 6.4: P-box for a_1 to a_5 obtained from the respective UMs.

$\pi_{\boldsymbol{a}}^5$. The resulting intervals constitute the updated hyper-rectangular set E defined by each of the resulting UM to which numerical results are presented in Table 6.6 in Section 6.3.3. Let the UM determined from $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$ given $\pi_{\boldsymbol{a}}^1$ be denoted as UM_{y0}^1, while that given $\pi_{\boldsymbol{a}}^5$ be denoted as UM_{y0}^2.

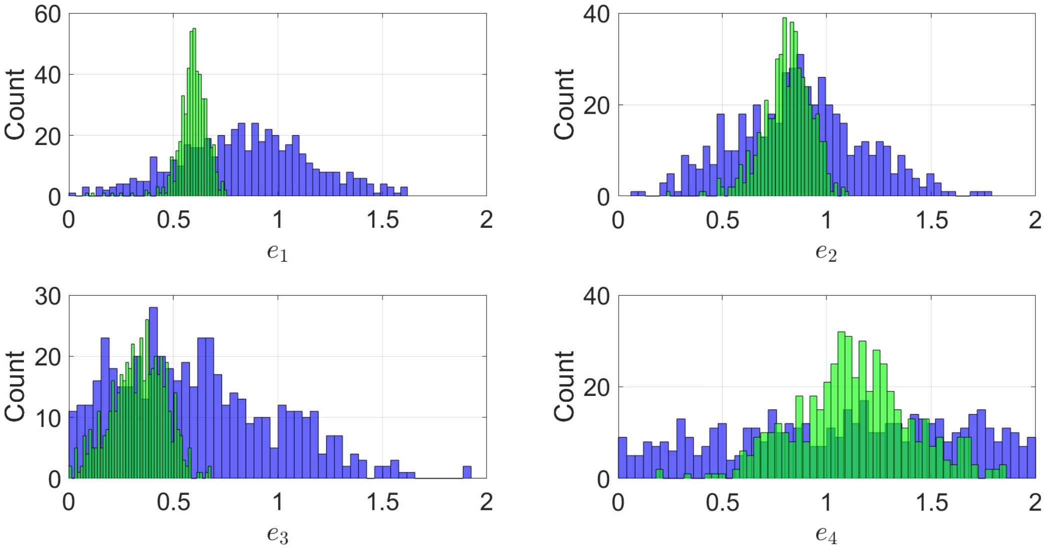


Figure 6.5: Histograms for e_1 to e_4 obtained from $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$ given $\pi^1_{\boldsymbol{a}}$ (in blue) and $\pi^5_{\boldsymbol{a}}$ (in green).

To verify the calibration results, $N = 500$ samples are generated from the hyper-rectangle defined by the bounds of the respective distribution parameters of $\pi_{\boldsymbol{a}}$ and e_{i_e} according to the respective UMs. For each sample realization from this hyper-rectangle, 100 model outputs of $\hat{y}$ is obtained for $t \in [0, 5]$ s. This is done by generating 100 realizations of $\boldsymbol{a}$ from $\pi_{\boldsymbol{a}}$, given the distribution parameters from the hyper-rectangle sample, whilst keeping $\boldsymbol{e}$ fixed. This yields a $N_t \times 100 \times N_s$ array of data output of $\hat{y}$ for each UM whose results are plotted in Figure 6.6. From the figure, it can be observed that the model output bands of UM^1_{y0} (in blue) and UM^2_{y0} (in green) generally encompasses D_1 (in red) which indicates that the model calibration procedure, via Bayesian model updating, was done satisfactorily.

To further substantiate this, P-boxes of the calibrated model output of each UM are constructed at $t = \{0.5, 1.0, 2.0, 3.0, 4.0, 5.0\}$ s. Each P-box describes the extreme bounds of the distribution of the N ECDFs whereby each ECDF comprises of the 100 model output values at t. Figure 6.7 presents the resulting P-boxes from UM^1_{y0} (in blue) and UM^2_{y0} (in green) at each chosen t. From the plots, it can be seen that the ECDF of D_1 (in red) at any given t is generally enclosed within the P-boxes. Furthermore, it can be observed that the shape of both P-boxes generally follow the shape profile of the ECDF of D_1 which indicates a good degree of fit by both UMs.

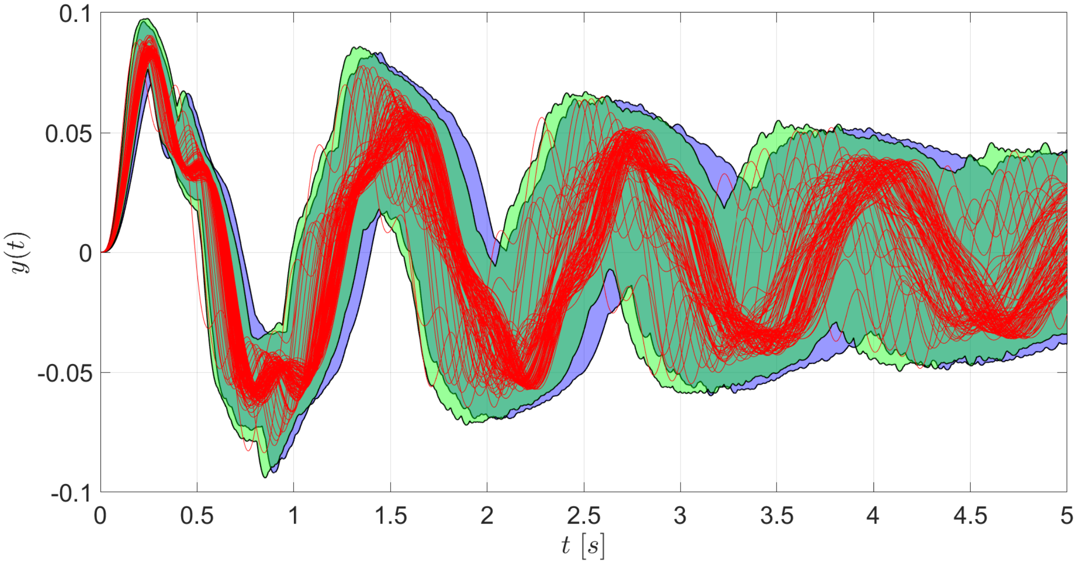


Figure 6.6: Output band from $\hat{y}$ according to UM^1_{y0} (in blue) and UM^2_{y0} (in green) along with the data sequence D_1 (in red) after calibration.

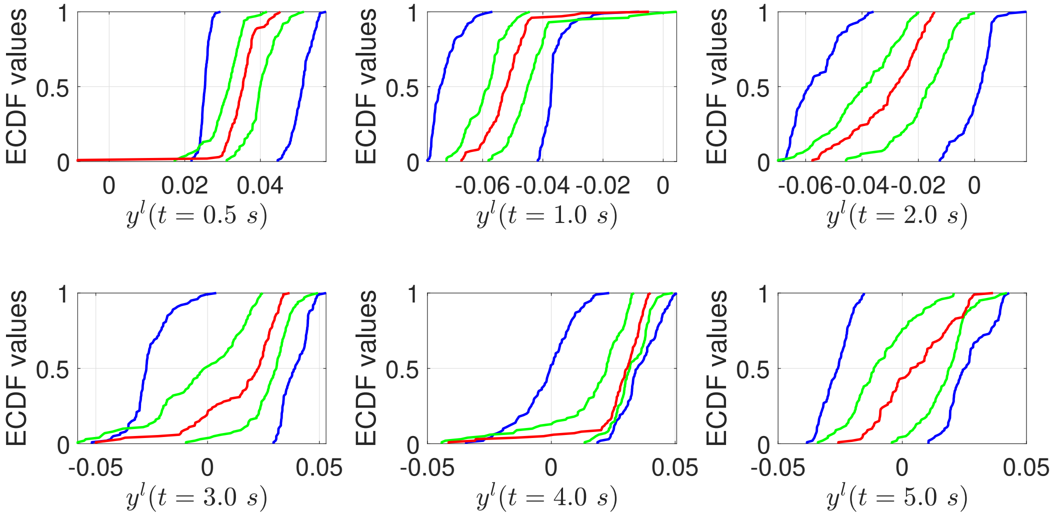


Figure 6.7: P-boxes of the model output from $\hat{y}$ obtained from UM^1_{y0} (in blue) and UM^2_{y0} (in green) at various time slices $t = \{0.5, 1.0, 2.0, 3.0, 4.0, 5.0\}$ s. The red ECDF denotes the distribution of the data D_1.

6.2.5 Further discussions

Figure 6.4 shows that the P-boxes obtained for a_1 to a_5 according to UM_{y0}^1 are generally wider than those according to UM_{y0}^2. This indicates a higher degree of uncertainty on the true distribution of all a_{i_a} by UM_{y0}^1 which makes it less informative in identifying the true $\pi_{\boldsymbol{a}}$ compared to UM_{y0}^2. In addition, the P-boxes for a_2 to a_5 obtained by UM_{y0}^2 are generally enclosed within those of UM_{y0}^1 which suggests that the true CDF defined by $\pi_{\boldsymbol{a}}$ could lie within the P-box defined by UM_{y0}^2.

The intervals obtained from the posterior distributions for e_1 to e_4 through the model based on Beta distributions is much wider compared to the UQ model obtained through the SRV based approach as shown in Figure 6.5. This further highlights the less-informative nature of UM_{y0}^1, especially for the case of e_3 and e_4. In addition, the posteriors obtained through the SRV based approach generally show a smaller spread about its means and that such approach is able to identify the epistemic parameters much more effectively given that the peaks are more pronounced. This leads to the uncertainty bounds of $\boldsymbol{e}$ according to UM_{y0}^2 being significantly narrower such that they are generally enclosed within that according to UM_{y0}^1 as seen from Table 6.6 in Section 6.3.3.

Figure 6.6 shows that the output bands of $\hat{y}$ obtained from both UMs follow the trend defined by D_1. However, from Figure 6.7, it is observed that the P-boxes obtained by UM_{y0}^2 have much tighter bounds compared to UM_{y0}^1 whilst still enclosing the ECDF of D_1. This is attributed to the P-box of the $\boldsymbol{a}$ and the bounds on $\boldsymbol{e}$ being narrower for UM_{y0}^2 than that for UM_{y0}^1. This results in the former yielding a significantly better degree of fit over D_1 than the latter. From the results, it can be concluded that UM_{y0}^1 is much more conservative compared to UM_{y0}^2 in modelling the uncertainty of $\pi_{\boldsymbol{a}}$ and $\boldsymbol{e}$.

6.3 Task 2: Uncertainty reduction

The objective of this task is to identify the epistemic parameters which have more predictive capability and improve the UM. This is achieved by performing a sensitivity analysis for the epistemic model parameters and the subsequent refinement of the epistemic space.

6.3.1 Sensitivity analysis

In this analysis, the epistemic uncertainties are ranked according to their ability to improve the predictive ability of the computational model of the subsystem. This predictive ability is quantified through the volume metric Ω defined as:

$$\Omega = \sum_{i_t=1}^{n_t} \rho_{i_t} \cdot \Delta_{i_t} \tag{6.11}$$

ρ_{i_t} is the area of the P-box at time-slice i_t, Δ_{i_t} is the time-step between time-slice $i_t - 1$ and i_t, and n_t is the total of time-slices used for the computation. For the computation, the 6 time-slices, which were used for the illustration of the P-boxes in Figure 6.7, are considered.

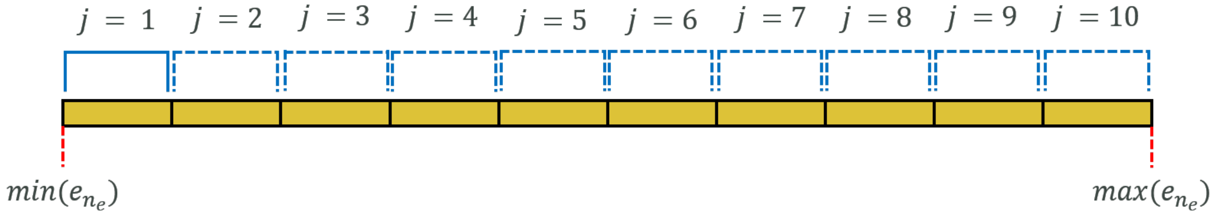


Figure 6.8: Illustration to the approach in identifying the maximum reduction of Ω from the pinching of e_{i_e}.

In order to rank the the epistemic parameters according to their respective sensitivity, an adaptive pinching method based on [36, 34] is proposed to provide a non-empirical approach to determine the pinched bounds of a chosen epistemic parameter which yields the greatest reduction in the value of Ω. To provide a visual understanding of the proposed method, a schematic diagram is provided in Figure 6.8. The procedure is as follows: For a given i_e, the uncertainty space of e_{i_e} is reduced by 90 %. This is done whilst keeping the uncertainty space of the remaining 3 epistemic model parameters untouched. For a given e_{i_e}, its bounds would first be divided into 10 equally-spaced units. Next, at iteration $j = 1$, a segment of bin length of 1 unit will be used to isolate the region of the epistemic space defined by the lower and upper bounds of the first bin. This isolated region serves as the reduced (or "pinched") space. From there, the corresponding realizations of $\{\boldsymbol{a}, \boldsymbol{e}\}$ from the UM, whose e_{i_e} value falls outside the bounds of the reduced epistemic space, is discarded. When this is done, the reduced volume Ω_p is computed again via Eq. (6.11). After this is done, the segment shifts by 1 unit to the right as shown in Figure 6.8 and this initiates iteration $j = 2$ where the above procedure is repeated all the way to iteration $j = 10$. This procedure is done for e_1 to e_4. As an illustrative example, the results of the reduced volume Ω_p for the respective iteration j for each e_{i_e} according to UM^1_{y0} are presented in Figure 6.9. From the figure, the minimum value of Ω_p for each e_{i_e} is determined and the sensitivity index is computed:

$$S = 1 - \frac{\Omega_p}{\Omega_0} \tag{6.12}$$

where Ω_0 is the initial volume before pinching. This sensitivity metric would then be used to rank e_1 to e_4 to which the results according to the respective UMs are shown in Table 6.3.

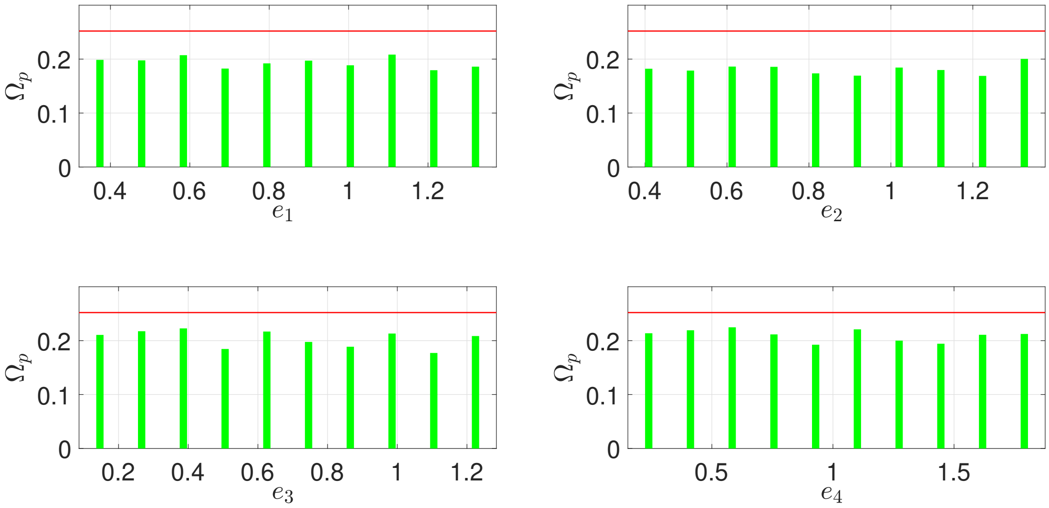


Figure 6.9: Results of Ω_p for different pinched intervals for e_1 to e_4 according to UM^1_{y0}. The red line denotes the initial volume $\Omega = 0.2521$, while the green bars represent the resulting Ω_p.

Rank	Uncertainty model UM^1_{y0}			Uncertainty model UM^2_{y0}		
	Parameter	Pinched bounds	S	Parameter	Pinched bounds	S
1	e_2	$[1.1729, 1.2748]$	0.3300	e_2	$[0.8677, 0.9117]$	0.4542
2	e_3	$[1.0449, 1.1647]$	0.2972	e_3	$[0.3595, 0.4072]$	0.3240
3	e_1	$[1.1619, 1.2670]$	0.2882	e_1	$[0.5715, 0.5987]$	0.2654
4	e_4	$[0.8425, 1.0148]$	0.2272	e_4	$[0.8242, 0.9320]$	0.2556

Table 6.3: The ranking order of the epistemic model parameters based on their respective sensitivity index for the respective UMs.

Based on the results provided in Table 6.3, it is observed that e_4 is ranked the lowest in sensitivity according to both UMs. This implies that it is impossible to improve the knowledge on e_4 with the available model and data, making impossible to extract or infer information on its true value, thus contributing the highest degree of non reducible epistemic uncertainty in the calibration of the UM. For this reason, the first refinement request to the challenge host was made for the lower bound of e_4, given the heavier left tail as seen from its histogram obtained via the Distribution-free approach (in green) in Figure 6.5. Following this, a second round of sensitivity analysis was performed following the approach outlined above and accounting for the given refined bounds of e_4 and the results are presented in Table 6.4. From the results, e_3 is consistently ranked within the bottom 2 according to both UMs which suggests that e_3 is the least informative parameter after e_4. Hence, the second refinement request was made for the lower bound of e_3 given the lack of such information according to both UMs as illustrated in Figure 6.5. The resulting epistemic space, with the refined e_3 and e_4 bounds, constitutes the hyper-rectangle epistemic space denoted as E_1.

Rank	Uncertainty model UM^1_{y0}			Uncertainty model UM^2_{y0}		
	Parameter	Pinched bounds	S	Parameter	Pinched bounds	S
1	e_4	$[1.0224, 1.0575]$	0.5200	e_2	$[0.6036, 0.6476]$	0.5999
2	e_2	$[0.4652, 0.5625]$	0.5114	e_4	$[1.1276, 1.1632]$	0.4128
3	e_1	$[0.6333, 0.7367]$	0.5056	e_3	$[0.2166, 0.2642]$	0.4024
4	e_3	$[0.9227, 1.0393]$	0.4282	e_1	$[0.5715, 0.5988]$	0.3821

Table 6.4: The ranking order of the epistemic model parameters based on their respective sensitivity index for the respective UMs accounting for the refined bounds for e_4.

6.3.2 Updated Uncertainty models

A second round of Bayesian model updating is performed with the bounds of the Uniform priors for the respective epistemic parameters defined by the hyper-rectangle E_1. The approach follows that outlined in Sections 6.2.2 and 6.2.3 from which UM^1_{y1} and UM^2_{y1} are obtained respectively. From which, the corresponding numerical results of the updated bounds for each e_{i_e} according to UM^1_{y1} and UM^2_{y1} are presented in Table 6.6 in Section 6.3.3. Finally, a sensitivity analysis was done again following the methodology presented in Section 6.3.1 and the results are summarized in Table 6.5. From the table, it can be observed that the sensitivity ranking of each e_{i_e} is the same as that in Table 6.3 for the respective UMs.

6.3.3 Results and discussions

The resulting model output of the response plot according to UM^1_{y1} and UM^2_{y1} are illustrated in Figure 6.10. From the figure, it can be observed that the response plots according to both UMs are well-fitted against D_1. However, such fitting is significantly tighter for the case of UM^2_{y1} as seen in Figure 6.10 and this is supported by Figure 6.11 where it can

Rank	Uncertainty model UM_{y1}^1			Uncertainty model UM_{y1}^2		
	Parameter	Pinched bounds	S	Parameter	Pinched bounds	S
1	e_2	$[0.4447, 0.5340]$	0.4095	e_2	$[0.9514, 0.9726]$	0.3610
2	e_1	$[0.9500, 1.0413]$	0.2960	e_3	$[0.5675, 0.6173]$	0.3245
3	e_4	$[0.9274, 0.9585]$	0.2385	e_1	$[0.5550, 0.5725]$	0.2981
4	e_3	$[0.3333, 0.3590]$	0.2379	e_4	$[1.0225, 1.0584]$	0.2782

Table 6.5: The ranking order of the epistemic model parameters based on their respective sensitivity index for the respective UMs accounting for the refined space E_1.

also be seen that the resulting P-boxes of the response plot across all chosen time-slices t are significantly narrower compared to UM_{y1}^1 whilst enclosing the ECDF for D_1. This observation is consistent to that discussed in Section 6.2.5 and concludes that the response plot according to UM_{y1}^2 is more representative of D_1.

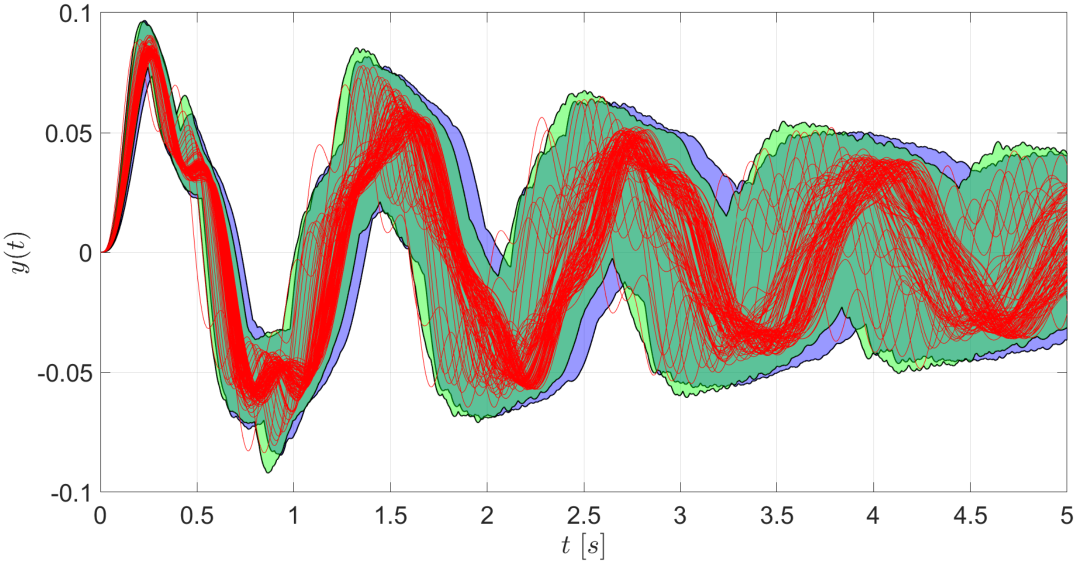


Figure 6.10: Output band from $\hat{y}$ according to UM_{y1}^1 (in blue) and UM_{y1}^2 (in green) along with the data sequence D_1 (in red) after calibration.

Figure 6.12 illustrates the resulting P-boxes quantifying the uncertainty over the marginal distributions of $\pi_{\boldsymbol{a}}$ by the respective UMs. From the figure, it can be seen that the P-boxes according to UM_{y1}^2 is significantly narrower and enclosed within that according to UM_{y1}^1 which verifies that the true marginal distributions of $\pi_{\boldsymbol{a}}$ could lie within the P-boxes defined by UM_{y1}^2. The results by UM_{y1}^1 once again highlights its conservative nature in its uncertainty over $\pi_{\boldsymbol{a}}$ compared to UM_{y1}^2.

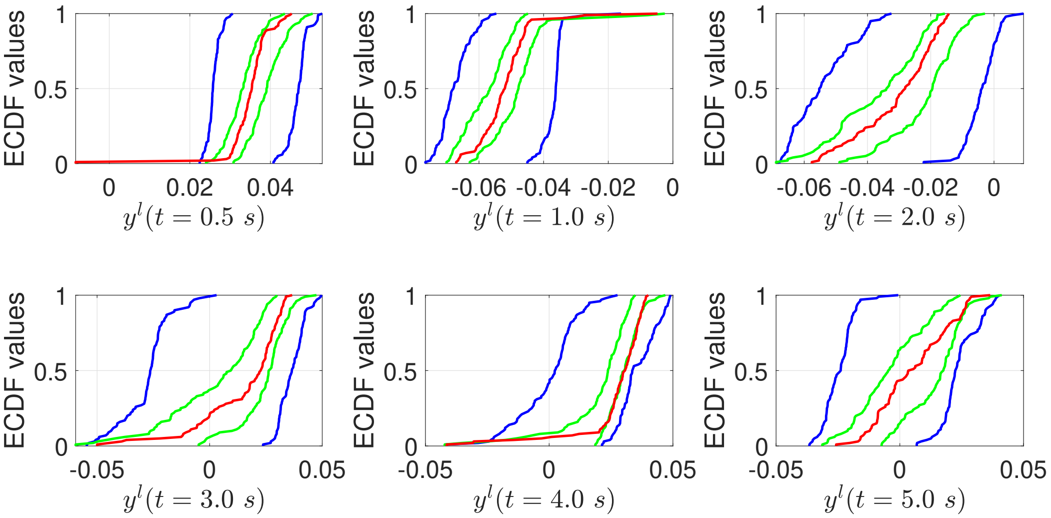


Figure 6.11: P-boxes of the model output from $\hat{y}$ obtained from UM^1_{y1} (in blue) and UM^2_{y1} (in green) at various time slices $t = \{0.5, 1.0, 2.0, 3.0, 4.0, 5.0\}\ s$. The red ECDF denotes the distribution of the data D_1.

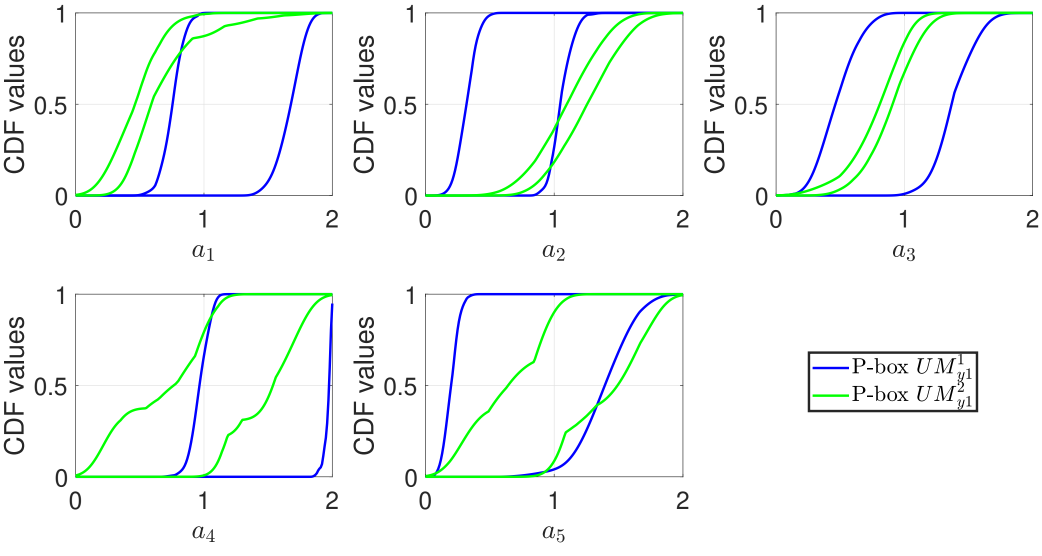


Figure 6.12: P-box for a_1 to a_5 obtained from the respective UMs.

Finally, results from Table 6.6 show that the uncertainty bounds over $\boldsymbol{e}$ according to UM^2_{y1} is significantly narrower and generally enclosed within that according to UM^1_{y1}. This result is supported by Figure 6.13 where it can be seen that the resulting histograms of the epistemic parameters obtained from $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$ given $\pi_{\boldsymbol{a}}^5$ are consistently narrower than that obtained from $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$ given $\pi_{\boldsymbol{a}}^1$. This verifies the results obtained by UM_{y1}^2 which further highlights its informative nature over UM_{y1}^1. For this reason, UM_{y1}^2 is chosen to address the subsequent tasks of the challenge.

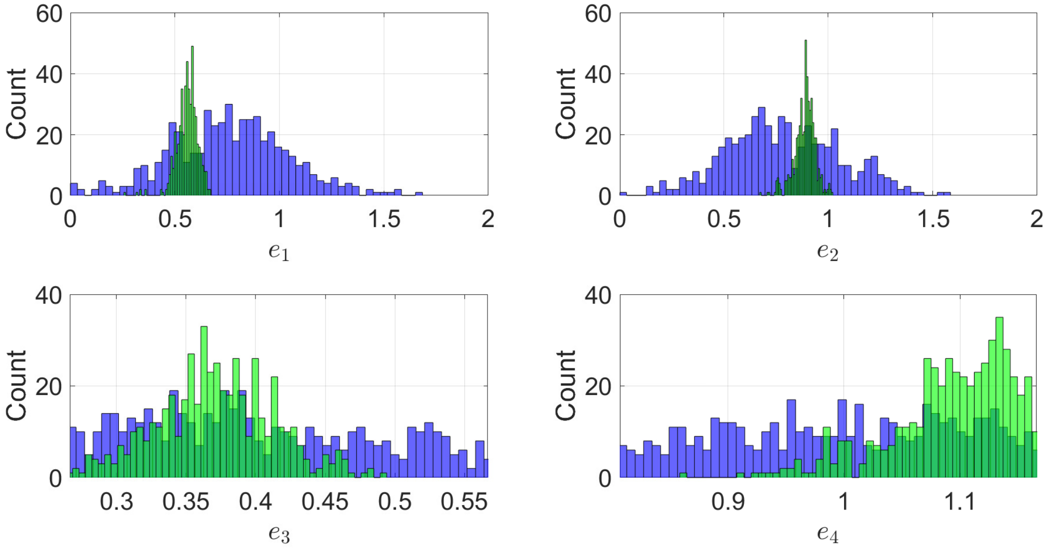


Figure 6.13: Histograms for e_1 to e_4 obtained from $P(\boldsymbol{\theta}|\boldsymbol{D}, M)$ given $\pi_{\boldsymbol{a}}^1$ (in blue) and $\pi_{\boldsymbol{a}}^5$ (in green).

Uncertainty model	e_1	e_2	e_3	e_4
UM_{y0}^1	$[0.3182, 1.3787]$	$[0.3574, 1.3771]$	$[0.0827, 1.2870]$	$[0.1486, 1.8828]$
UM_{y0}^2	$[0.4351, 0.7082]$	$[0.5583, 1.0000]$	$[0.0721, 0.5511]$	$[0.6066, 1.6893]$
UM_{y1}^1	$[0.3097, 1.2306]$	$[0.3522, 1.2487]$	$[0.2819, 0.5400]$	$[0.8337, 1.1461]$
UM_{y1}^2	$[0.4674, 0.6433]$	$[0.7607, 0.9736]$	$[0.2865, 0.4583]$	$[0.9627, 1.1664]$

Table 6.6: Updated epistemic space E for e_1 to e_4 according to the respective UMs.

6.4 Numerical implementation and computational time

In addressing the challenge tasks presented in the chapter, the adopted algorithms are mainly based on random sampling and stochastic algorithms. Thus, the execution time fluctuates significantly due to the inherent randomness depending on the uncontrolled conditions such as starting samples, the evolution of the samples, etc. However, it needs to be noted that the random seed has not been fixed to allow for generality of the implementation and solution. Therefore, the chapter only provides the approximate timing as the performance indicator: 1) the TEMCMC which takes between 5 to 8 hours of sampling time; and 2) the adaptive pinching approach which involves less than a minute of computation time.

Finally, it needs to be highlighted that the computational times stated for the respective tools are also dependent on the computational efficiency of the high-performance CPUs which are used. These estimated timings may differ between different CPUs of different specifications.

6.5 Chapter conclusion

The chapter has presented and demonstrated the different techniques implemented to address Tasks 1 and 2 of the NASA-Langley Uncertainty Quantification Challenge.

In Task 1, the Bayesian model updating technique is used to calibrate the Uncertainty Model by performing a probabilistic model update, via the Transitional Ensemble Markov Chain Monte Carlo sampler, on both the distribution of the aleatory parameters as well as the epistemic parameters. To provide a thorough comparison between different set-ups, 2 different Uncertainty Models have been proposed and analysed. Each of these models adopt a different choice of joint distribution function for the aleatory space (i.e. the Beta distribution vs the Staircase Density Functions), stochastic distance metric (i.e. Wasserstein distance vs Bhattacharyya distance), and data type (i.e. frequency domain vs time domain). Based on the results presented in Sections 6.2.4 and 6.3.3, it can be concluded that the use of the Distribution-free approach provided more informative results on the bounds of the distribution of the aleatory parameters and the epistemic parameters.

In Task 2, an adaptive pinching analysis was proposed to perform the sensitivity analysis on the epistemic parameters, providing an efficient way of identifying the largest possible reduction of the proposed volume metric by the single pinched component of the epistemic space as well as the reduced bounds of the respective epistemic components. In doing so, it allows for a systematic, non-empirical way to justify the pinched bounds and ensure that all regions of the individual components of the epistemic space are accounted for in the investigation of their respective effect on the 2 aforementioned quantities.

Chapter 7

Sequential Ensemble Monte Carlo: A Robust and Efficient sampler for On-line Bayesian Model Updating

Abstract

Several on-line identification approaches are used for the "real-time" identification of parameter(s) of engineering systems and structures via Bayesian inference. For this purpose, a robust and "tune-free" sampler is proposed and is referred to as the Sequential Ensemble Monte Carlo sampler. The chapter presents an introduction to its theoretical framework which seeks to extend an implementation of the Sequential Monte Carlo sampler towards the identification of time-varying parameters which can be assumed constant whilst each set of data collected, but may vary across the different sequences of data-sets. The proposed approach involves the implementation of the Affine-invariant Ensemble sampler in place of the Metropolis-Hastings sampler to update the samples. An adaptive-tuning algorithm is also proposed to automatically tune the step-size of the Affine-invariant ensemble sampler which, in turn, controls the acceptance rate of the samples across iterations. This is complemented with a numerical investigation that provides evidence of the existence of an inherent minimum value and maximum value on the acceptance rate, making the proposed sampler robust by design.

7.1 Background

In recent years, on-line learning has garnered significant attention for the purpose of parameter identification of engineering systems. On-line parameter identification involves the learning and estimation of the parameter(s) of interest through distinct data-sets which are obtained sequentially, contrary to the batch learning approach which requires the availability of the entire data-set to produce estimates [276]. In particular, on-line parameter identification is of value in situations when data is obtained over a period of time. This allows for real-time parameter identification, making it a practical approach for investigating the performance of engineering systems under operating conditions. For example, on-line parameter identification has been applied to identify: moveable mass positions within a 2DoF shear frame [264, 1]; mistuning parameters of rotating blisks [277]; modal parameters of a vehicle motion modes to investigate its dominant motion-mode [278]; structural parameters of a nonlinear structural system to update its dynamical response model [279]; terrain parameters of the Wheel-Terrain model for wheeled motion control of mobile robots [280]; and structural parameters of smart building structures for real-time damage detection [281].

The parameters identified through on-line learning are often time-invariant, see e.g. [243, 277, 278]. However, in many engineering applications, these parameters vary with time [282] to which examples include: 1) structural modal parameters to study the dynamic response of structures [221, 222]; 2) fatigue cracking parameters [223]; 3) stress data for performance prediction of steel bridges [283]; 4) localized impact damage in composite panels based on sensor data [284]; and 5) earthquake ground motion parameters using generalized Kalman filter and structural absolute accelerations data [285].

Currently, within the time-domain applications, model parameters and system states (i.e. displacements and velocities) are estimated via sequential Bayesian inference using Kalman filters [286, 287], Gaussian Sum filters [288, 289], and Particle filters [290, 291]. Kalman filters are computationally less-expensive compared to other filtering techniques. However, they are designed specifically to deal with problems involving a linear dynamical system and a Gaussian "noise" [292]. The Extended Kalman filter [293, 294] and the Unscented Kalman filter [279, 286] extended the approach towards non-linear dynamical systems and non-Gaussian "noise". For instance, Gaussian Sum filters utilises weighted Gaussian Models to approximate the predictive and posterior PDFs [289]. Thus, it does not require the analytical form of the aforementioned PDFs. However, like the Kalman filter, Gaussian Sum filters become ineffective when the dynamical system set-up becomes highly non-linear [295].

Particle filters are applicable in both linear and non-linear dynamical systems and do not assume the form of the “noise” [296]. However, they are computationally expensive and perform poorly even with moderately high number of parameters (i.e. above 18 dimensions) [210, 46]. For more information pertaining to the above approaches, the reader is referred to the recent review paper by [297].

The focus of the chapter is on the SMC sampler whose concept was introduced in Chapter 3. Currently, it has been implemented under off-line (i.e. batch) and on-line settings towards the parameter identification for numerous set-ups including: 1) non-linear time-series model; 2) non-linear state-space models; and 3) high-dimensional target distributions [178, 298]. At present, the SMC sampler has not yet been implemented to identify model parameters that are considered constant while a sequence of observations is obtained but may vary across the different sequences of such observations. Under such settings, the posteriors of such parameters are assumed to be static within an observation sequence from which the SMC sampler is implemented to generate samples to provide estimates on the parameters at that given time step sequence. The posteriors, and therefore the estimates on the parameters, may vary between the different sequences of such observations. It needs to be emphasised that such problem that is being investigated in the dissertation (i.e. see Chapter 8) is different to those that can be addressed with the filtering methods such as the Kalman filter and the Particle filter. Moreover, there exists the following short-comings in the SMC sampling algorithm: 1) the choice of proposal distribution can significantly affect the sampler results [52]; 2) it is computationally inefficient because of the number of parameters required to compute and tune [299]; and 3) the moderation of acceptance rates of its samples has not been investigated in detail.

To address these short-comings, a robust and “tune-free” Sequential Ensemble Monte Carlo (SEMC) sampler is proposed based on the use of the AIES in place of the MH sampler for the MCMC step [300]. The AIES has been proven to be robust in recent implementations for reliability analysis using Subset Simulation [301], and for model updating via the TEMCMC sampler [101, 264] thanks to its capability of sampling from highly-skewed and anisotropic distributions (i.e. see Chapter 4 of the dissertation). In addition, an adaptive-tuning algorithm, inspired from the work by [169], is developed to provide a robust mechanism to ensure that the acceptance rate values have achieved convergence before moving on to the next sampling iteration. Details on the adaptive-tuning algorithm are provided in Section 7.4.1. Numerical investigation has proven the existence of inherent bounds in the acceptance rates values and shown in Section 7.4.2. To the best of the author’s knowledge, such investigation and analysis is yet to be presented in existing literature.

7.2 Sequential Bayesian inference

Sequential Bayesian inference is a popular technique to address inverse problems and infer time-varying model parameter(s) under uncertainty [37, 38]. Its mathematical formulation extends from the Bayesian inference framework implemented to infer time-invariant model parameter(s) as outlined in Chapter 2. As stated in Section 7.1, the interest here is in the use of the sequential Bayesian inference framework towards inferring the time-varying parameter(s) whose posterior is consequently static for a given observation sequence but varying over time between the different observation sequences. To reflect such time-varying aspects, $\boldsymbol{\theta}^s$ is defined as the parameter(s) at the s^{th} time step sequence whose instantaneous (i.e. static) posterior can be expressed as $P(\boldsymbol{\theta}^s|\boldsymbol{D}^{1:s}, M)$, while $\boldsymbol{D}^{1:s} = \{\boldsymbol{D}^1, \ldots, \boldsymbol{D}^s\}$ denotes the stream of data-set $\boldsymbol{D}$ obtained sequentially up to the s^{th} time step sequence in an on-line manner. An essential requirement in the inference of $\boldsymbol{\theta}^s$ is the underlying Markov kernel $T(\boldsymbol{\theta}^{s+1}|\boldsymbol{\theta}^s)$ that describes the evolution from $\boldsymbol{\theta}^s$ to $\boldsymbol{\theta}^{s+1}$ and can be expressed as [283]:

$$T(\boldsymbol{\theta}^{s+1}|\boldsymbol{\theta}^s) : \boldsymbol{\theta}^{s+1} = \Gamma(\boldsymbol{\theta}^s) + \nu_{\boldsymbol{\theta}} \tag{7.1}$$

where $\Gamma(\bullet)$ is the nominal evolution model and $\nu_{\boldsymbol{\theta}}$ is the process "noise". In this work, we shall assume that $\nu_{\boldsymbol{\theta}}$ follows a zero-mean Normal distribution with fixed standard deviation σ_ν (see e.g. [283, 302, 303]).

The sequential Bayesian inference procedure can be summarised as follows [282]: At time step sequence $s = 1$, the posterior $P(\boldsymbol{\theta}^s|\boldsymbol{D}^{1:s}, M)$ is defined (see Eq. (2.8)). Following which, the predictive distribution $P(\boldsymbol{\theta}^{s+1}|\boldsymbol{D}^{1:s}, M)$ is computed [296]:

$$P(\boldsymbol{\theta}^{s+1}|\boldsymbol{D}^{1:s}, M) = \int T(\boldsymbol{\theta}^{s+1}|\boldsymbol{\theta}^s) \cdot P(\boldsymbol{\theta}^s|\boldsymbol{D}^{1:s}, M) \cdot d\boldsymbol{\theta}^s \tag{7.2}$$

The predictive distribution $P(\boldsymbol{\theta}^{s+1}|\boldsymbol{D}^{1:s}, M)$ describes our prediction of $\boldsymbol{\theta}^{s+1}$ before observing the data $\boldsymbol{D}^{s+1}$ to be obtained in time step sequence number $s + 1$. In this regard, $P(\boldsymbol{\theta}^{s+1}|\boldsymbol{D}^{1:s}, M)$ is set as the new prior to be updated and the process is repeated for time step sequence number $s + 1$ until the terminal sequence s_{end}.

To sample sequentially from the given static posterior $P(\boldsymbol{\theta}^s|\boldsymbol{D}^{1:s}, M)$, the SMC-MH sampler is implemented as follows [181]: At iteration $s = 0$, sampling algorithm is initialized by generating N samples generated from the prior $P(\boldsymbol{\theta}^s)$. Next, the normalised weights of each

i^{th} sample $\hat{w}_i^s$ is computed following Eq. (3.20). Following which, the algorithm proceeds to the updating step where N single-step Markov chains are initiated, each starting from sample $\boldsymbol{\theta}_i^s$ obtained using weighted resampling (with replacement) according to $\hat{w}_i^s$. The Metropolis-Hastings (MH) approach is then adopted to generate 1 sample from each Markov chains [50]. The candidate samples are generated from a Normal proposal distribution $q(\boldsymbol{\theta}_i^{s*}|\boldsymbol{\theta}_i^s)$ with mean $\bar{\boldsymbol{\theta}}^s$ and covariance matrix $\boldsymbol{\Sigma}^s$ computed using Eq. (3.9) and Eq. (3.8) respectively. From there, the candidate sample is accepted or rejected using Algorithm 2. After which, the updated samples $\boldsymbol{\theta}_i^s$ are then passed through the Markov kernel $T(\boldsymbol{\theta}^{s+1}|\boldsymbol{\theta}^s)$ to generate $\boldsymbol{\theta}_i^{s+1}$ (i.e. the predictive samples). Finally, the predictive distribution function is estimated using a Kernel Density Estimate in the form of:

$$P(\boldsymbol{\theta}^{s+1}|\boldsymbol{D}^s, M) \approx \frac{1}{N}\sum_{i=1}^{N} K\left(\frac{\boldsymbol{\theta} - \boldsymbol{\theta}_i^{s+1}}{\boldsymbol{h}}\right) \tag{7.3}$$

where $\boldsymbol{\theta}$ is a random variable, $K(\bullet)$ is the Kernel smoothing function which is set as the standard Normal distribution, and $\boldsymbol{h}$ is the N_d-by-N_d diagonal bandwidth matrix where N_d is the number of inferred parameter(s). Each diagonal element of the bandwidth matrix h_d, for $d = 1, \ldots, N_d$ is computed using the Silverman's Rule of Thumb [304]:

$$h_d = \tilde{\sigma}_d \cdot \left[\frac{4}{N \cdot (d+2)}\right]^{\frac{1}{(d+4)}} \tag{7.4}$$

where $\hat{\sigma}_d$ is the standard deviation of the d^{th} component of $\boldsymbol{\theta}^{s+1}$. The algorithm proceeds to the next time step $s = s + 1$ if data are available by setting $\boldsymbol{\theta}_i^{s+1}$ as the new prior samples and $P(\boldsymbol{\theta}^{s+1}|\boldsymbol{D}^{1:s}, M)$ as the new prior PDF. When no further data is obtained beyond that point, the algorithm terminates at time step $s = s_{end}$. A pseudo-algorithm of the sampling procedure by the SMC-MH sampler is presented in Algorithm 8.

Algorithm 8 SMC-MH sampler algorithm

```
procedure (Generate $N$ samples sequentially from $P(\boldsymbol{\theta}^s|\boldsymbol{D}^{1:s}, M)$)
    Set $s = 0$                                                        ▷ Initialise counter
    Draw initial $N$ sample set: $\boldsymbol{\theta}_i^s \sim P(\boldsymbol{\theta}|M)$        ▷ Generate samples prior
    while $s < s_{end}$ do                                             ▷ Main sampling loop
        Set $s = s + 1$
        Compute $\hat{w}_i^s$ using Eq. (3.20)
        for $i = 1 : N$ do                                 ▷ For each $i^{th}$ chain (MCMC step)
            Resample: $\boldsymbol{\theta}_i^s \sim \hat{w}_s^j$
            Draw candidate sample: $\boldsymbol{\theta}_i^{s*} \sim q(\boldsymbol{\theta}_i^{s*}|\boldsymbol{\theta}_i^s)$
            Accept/Reject $\boldsymbol{\theta}_i^{s*}$ with probability $\alpha_i$ using Algorithm 2
        end for
        Compute $\boldsymbol{\theta}_i^{s+1}$ using $T(\boldsymbol{\theta}^{s+1}|\boldsymbol{\theta}^s)$          ▷ Set as new prior samples
        Compute PDF of $P(\boldsymbol{\theta}_i^{s+1}|D^{1:s})$ using Eq. (7.3)          ▷ Set as new prior
        Compute $P(\boldsymbol{D}^{1:s}|M)$ using Eq. (4.1)
    end while
end procedure
```

7.3 Review of Sequential Monte Carlo

7.3.1 Advantages and limitations of current SMC sampler

One key advantage of the SMC sampler lies in its ability to sequentially compute the evidence term $P(\boldsymbol{D}^{1:s}|M)$ corresponding to the posterior $P(\boldsymbol{\theta}^s|\boldsymbol{D}^{1:s}, M)$ at any given s^{th} time step sequence [52, 305]. The metric $P(\boldsymbol{D}^{1:s}|M)$ also quantifies how well a given model class M describes the available set of data $\boldsymbol{D}^{1:s}$ as well as the time-evolution of $\boldsymbol{\theta}$. This makes the SMC sampler well-suited in addressing problems regarding the model class selection of M (see e.g. [306, 307, 305, 308, 309]). The evidence term $P(\boldsymbol{D}^{1:s}|M)$ can be estimated by the product of the mean of the nominal weights w_i^s at any given time step sequence $s \geq 1$ using Eq. (4.1).

Another characteristic of the SMC sampler is the flexibility in the choice of auxiliary parameters of the algorithm such as the scaling parameter of the covariance matrix and the Markov kernel of the time-varying parameter (i.e. see Eq. (7.1)) [310, 311]. This is because even if the samples do not follow the true distribution, the weighting process and the conditional resampling step (i.e. the Bootstrapping with replacement), with which the algorithm will correct and move the samples closer to its true distribution. To illustrate the resampling concept, a numerical example involving a a mixture of 2 bi-variate Gaussian distributions based on work in [51] is presented. Here, we shall consider a 2-dimensional Uniform prior such that: $P(\theta_1) \sim U(-2, 2)$ and $P(\theta_2) \sim U(-2, 2)$. The likelihood function is defined as a mixture of 2 bi-variate Gaussian functions with means centered about $\{\theta_1, \theta_2\} = \{0.5, 0.5\}$ and $\{-0.5, -0.5\}$ with covariance matrix $0.1 \cdot \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. As an example, the SMC-MH sampler is implemented with $N = 10000$ samples from which the prior samples with their associated normalised weights $\hat{w}_i^s$ are presented in Figure 7.1. In the figure, the samples which fall in the regions of the posterior indicated in yellow and light blue are resampled with a higher probability than those which fall outside the regions of the posterior indicated in dark blue. This allows for the Markov chains in the MCMC step to initiate within the support of the posterior and removes any need to consider the burn-in.

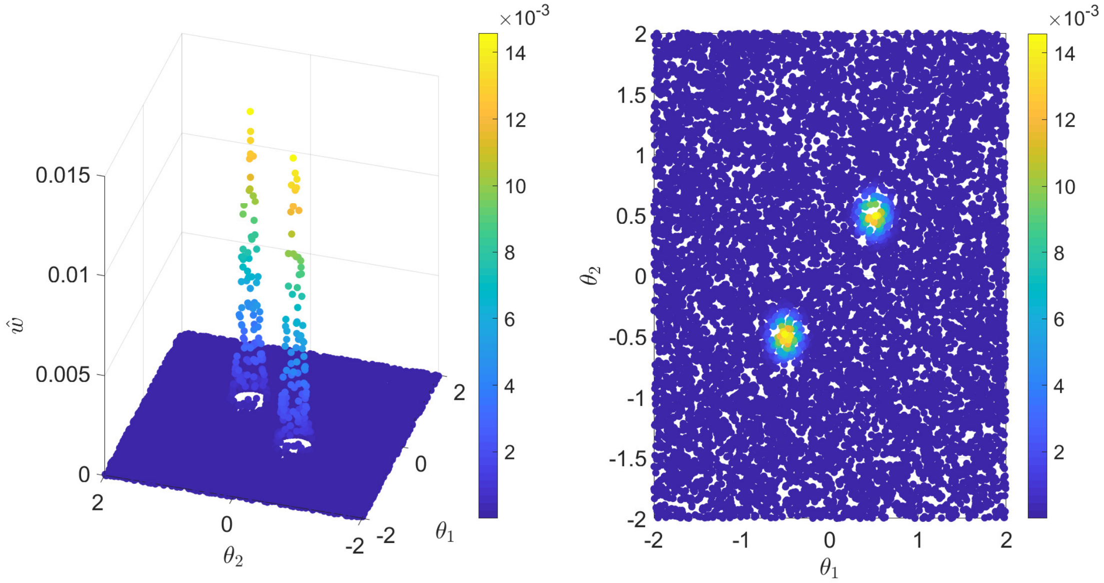


Figure 7.1: Scatterplot of the prior samples, along with their associated normalised weights $\hat{w}_i^s$, obtained from the posterior consisting of a mixture of two Gaussian distributions.

However, there are still limits to such flexibility as situations could arise whereby nearly all the samples could still fall outside the true distribution even after the conditional resampling step. This leads to nominal weights being close to 0 and the resampling step would fail to converge the samples to the true distribution [312].

One key problem in the SMC sampler is that there exists no universal choice of proposal distribution $q(\boldsymbol{\theta}_i^{s*}|\boldsymbol{\theta}_i^s)$ to generate candidate samples $\boldsymbol{\theta}_i^{s*}$ [52]. This creates significant degree of model uncertainty in deciding an appropriate distribution for $q(\boldsymbol{\theta}_i^{s*}|\boldsymbol{\theta}_i^s)$. In practice, a

Normal distribution is used to ease the computation of the acceptance ratio α_i due to the symmetric nature of the distribution [50, 233]. However, the ideal choice of $q(\boldsymbol{\theta}_i^{s*}|\boldsymbol{\theta}_i^{s})$ is one that is "optimally" scaled to the current posterior $P(\boldsymbol{\theta}^s|\boldsymbol{D}^{1:s}, M)$ [52, 233, 82]. Practically speaking, this would be difficult for two reasons: 1) there is a lack of knowledge over the analytical form of the true posterior itself; and 2) it is difficult to determine that "optimal" scale, especially in the case where the random-walk MH algorithm is used as the MCMC kernel (i.e. the SMC-MH sampler), although the problem can be addressed through the use of the independent MH MCMC kernel in [181] which eliminates the need for parameter tuning.

Another challenge is the computational cost of the SMC sampler. In particular, when the dimensionality of the problem increases, a higher computation cost is incurred in computing the covariance matrix of $q(\boldsymbol{\theta}_i^{s*}|\boldsymbol{\theta}_i^{s})$ at each s^{th} time step sequence, which can be the case for both the independent MH [181] and random-walk MH algorithms.

7.3.2 Sequential Monte Carlo variants

Numerous MCMC move kernels within the SMC sampler have recently been considered such as: 1) Particle Evolution Metropolis (i.e. PEM-SMC) [184, 313], 2) Gibbs sampler (i.e. SMC for Vector Auto-regressions with Stochastic Volatility (VAR-SV)) [314], and 3) target-invariant MCMC mutation kernel (i.e. SMC for high-dimensional inverse problems) [214].

The PEM-SMC sampler has demonstrated its strength in generating samplers more effectively from complex-shaped distributions, especially those with multiple peaks [184]. The sampler is also able to sample efficiently from moderate-dimensional posteriors (i.e. up to 30 dimensions) thanks to its effective way to explore the dimensional sample space and generate more candidate samples with a high probability content. This ensures a quick convergence of the samples towards the posterior distribution [184]. However, the effectiveness of the algorithm is subjected to the choice of the proposal distribution (i.e. the transfer probability distribution) which leads to the potential problem of model uncertainty.

The SMC sampler for VAR-SV has demonstrated its strength in tackling the problem of degeneracy effectively through the use of the Gibbs sampler which ensures that there is less repeated samples generated in the MCMC step. This increases the number of unique samples, thereby, allowing the sample space defined by the posterior to be well-explored [314]. In addition, it can be easily parallelised which allows for the rapid update of samples from one posterior to the next across iterations [314]. However, the sampler is limited by the short-coming of the Gibbs sampler which is the latter's dependency on the choice of an appropriate conditional distribution to represent the posterior [136]. Should the posterior be functionally complex, choosing such conditional distribution becomes non-trivial as highlighted in Chapter 3 of the dissertation [46].

The target-invariant MCMC mutation kernel is also quite robust thanks to its ability to address the issue of sample degeneracy effectively like the SMC for VAR-SV [214]. In addition, the algorithm ensures a relatively quick convergence rate of the samples to the posterior distribution through the adaptive tempering step which ensures a smooth transition from one posterior to the next between successive iterations [214]. This allows for the sampler to be applicable to cases with highly-dimensional and complex-shaped posteriors. However, the use of such MCMC move kernel introduces a relatively large number of auxiliary parameters to compute such as the tempering parameter and the Fourier coefficients which increases the computational cost of the algorithm [214].

For the work presented here, the AIES is implemented as the alternative MCMC kernel to the MH. The reasons behind such proposal are: 1) to exploit the strength of the AIES in its ability to sample from anisotropic and highly-skewed distributions (i.e. see Section 4.3.1 of the dissertation); and 2) the AIES can be parallelised. This gives rise to the SEMC sampler to which an additional feature proposed is the adaptive tuning algorithm which serves to control the acceptance rate of the sampler by tuning its step-size parameter in an automatic manner.

7.4 Sequential Ensemble Monte Carlo sampler

A key feature of the proposed SEMC sampler is in the implementation of the AIES in place of the MH sampler as the MCMC kernel to update the samples according to the current posterior $P(\boldsymbol{\theta}(t_s)|\boldsymbol{D}^{1:s}, M)$. Detailed mathematical descriptions on the AIES, as well as a discussion on its key advantages over the MH sampler, are provided in Section 4.3.1 of the dissertation.

7.4.1 Adaptive-tuning algorithm

The proposed adaptive-tuning algorithm is based on the work by [169] which serves 2 key purposes: 1) to adaptively tune and update the step-size u; and 2) to provide a mechanism to control the acceptance rates of the SEMC sampler such that they converge towards the user-defined target acceptance rates and fall within the optimal bounds of $[0.15, 0.50]$ suggested by [150].

The procedure undertaken by the algorithm is as follows: At $s = 1$, an initial step-size $u^{s=1}$ is defined by the user to which the recommended value is 2 [231, 239]. From this initial value, the nominal step-size u_{nom} is computed after the MCMC step following Eq. (4.16). If $u_{nom} > 1$, then $u^{s+1} = u_{nom}$. Otherwise, the algorithm sets $u^{s+1} = 1.01$ to ensure that the step-size would never be less than 1. This procedure is then repeated at the end of each iteration until the terminal iteration $s = s_{end}$.

To provide additional robustness to the SEMC sampler and assuring the acceptance rates converge towards the target acceptance value at every iteration, "virtual" iterations are introduced which involves the updating of the posterior samples with a series of repeated data $\boldsymbol{D}^s$. The termination criteria is defined whereby the acceptance rate values have converged. This is indicated when the difference in the acceptance rate values Δ_α between successive "virtual" iterations j falls within 10 % of the value of α_{tr} upon which the procedure ends. In doing so, it not only allows for the automatic tuning of the step-size u^s but also ensures that the acceptance rate values converge towards α_{tr} across all s, independent of the data provided.

7.4.2 Acceptance rates analysis

It needs to be noted that the target acceptance rate defined by Eq. (4.2) is based on the analysis that was done for the MH sampler. There is currently no literature that looks into the mathematical computation of the target acceptance rate for the AIES. While this is used as a general reference for the AIES, there exists a minimum acceptance rate value based on the analysis done in the context of the numerical example involving a mixture of 2 bi-variate Gaussian distributions [51] presented in Section 7.3.1. For this study, the distribution sequence P^j is constructed following Eq. (3.6) with the step size set at $\Delta\beta_s = 0.1$ across all time step sequences s, giving a total of 10 time step sequences. A sample size of 1000 samples is obtained from each P^j. In total, 6 different simulation runs are conducted with the corresponding acceptance rates $\alpha_{tr} = \{0.100, 0.283, 0.440, 0.800, 0.900, 1.000\}$ and the respective starting step-size values of $u^{s=1} = \{15, 15, 10, 3, 2, 2, 2\}$ to which the results are presented in Figure 7.2.

From the results, it can be seen that the acceptance rates never fall below 0.300 despite having a larger value of $u^{s=1}$ and a lower target acceptance rate values as seen in the cases of $\alpha_{tr} = 0.100$ and $\alpha_{tr} = 0.283$ The reason behind the existence of a minimum acceptance rate value is attributed to the resampling procedure by the SEMC algorithm which ensures that samples far from the posterior are eliminated while samples closer to or within the posterior are re-populated. As a result, a proportion of the samples will always be accepted, resulting in a non-zero acceptance rate. On the other hand, the upper bound of the acceptance rate is in practice always less than 1 given that for a non-Uniform distribution a proportion of samples are rejected by the sampling algorithm. In fact, in the cases where the chosen target acceptance rate values were 0.900 and 1.000, the acceptance rates across the time step sequences s never exceeded 0.850 as seen in Figure 7.2.

Hence, the existence of such bounds makes the algorithm particularly robust and generally applicable without the need of guessing a "good" acceptance rate. Instead, the acceptance rate can be used as a monitoring parameter of the performance of the algorithm. In fact, if the acceptance rate is high (e.g. above 0.850), it could indicate that the samples are stuck in a specific region of the posterior and the resulting sample distribution and its estimates may not be representative of the true posterior distribution. The acceptance rate bounds may depend on the dimension of the posterior distribution and the number of chains N_c used in the sampler which requires further investigation.

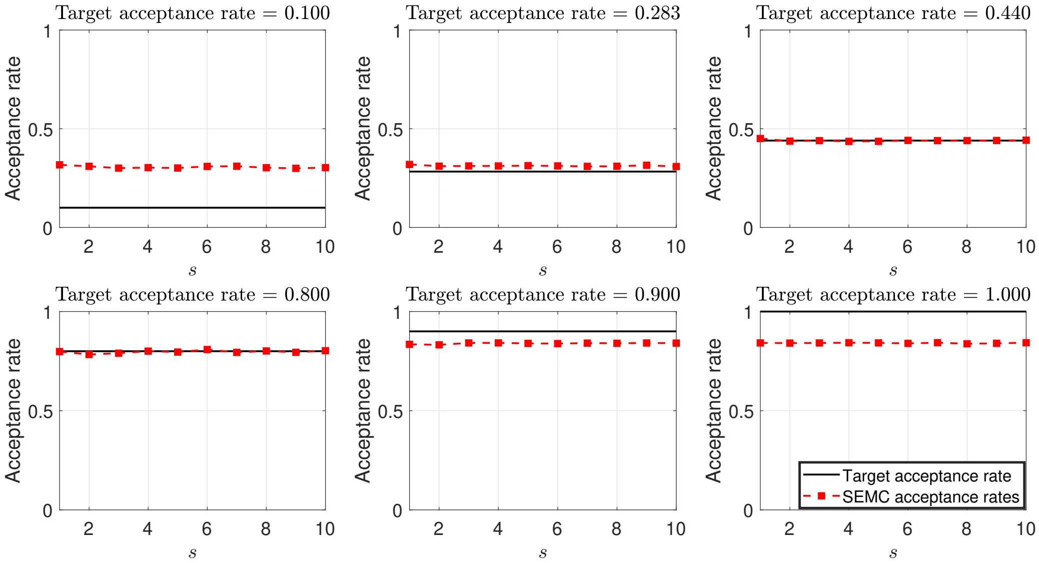


Figure 7.2: Results of the acceptance rates evolution by the SEMC sampler across s given the target acceptance rate values $\alpha_{tr} = \{0.100, 0.283, 0.440, 0.800, 0.900, 1.000\}$ with the respective starting step-size values $u^{s=1} = \{15, 15, 10, 3, 2, 2, 2\}$.

Without the need to decide on a "good" acceptance rate, complemented with the proposed adaptive-tuning algorithm involving the "virtual" iterations, these features highlight the robustness of the proposed SEMC sampler in that: 1) it is "tune-free" for the users in that they do not have to define an initial step-size value $u^{s=1}$; and 2) it is able to effectively control the acceptance rates across the different time step sequence s as illustrated from the results in Figure 7.2.

Algorithm 9 presents a summary of the SEMC sampling procedure. The resampling step in line 10 ensures that: 1) the Markov chains initiate with a higher probability from samples $\boldsymbol{\theta}_i^s$ with a higher w_i^s [51]; and 2) the final distribution of samples would be representative of the analytical distribution of $P(\boldsymbol{\theta}_i^s|D^{1:s}, M)$.

Algorithm 9 Proposed SEMC sampler algorithm

```
 1: procedure (Generate $N$ samples from $P(\boldsymbol{\theta}^s|\boldsymbol{D}^{1:s}, M)$)
 2:     Set $s = 0$                                              ▷ Initialise time step counter
 3:     Draw initial $N$ sample set: $\boldsymbol{\theta}_i^s \sim P(\boldsymbol{\theta}|M)$     ▷ Generate samples from prior
 4:     Set $u^{s+1} = 2$                                        ▷ Set initial value of step-size
 5:     while $s < s_{end}$ do                                   ▷ Loop over time steps
 6:         Set $s = s + 1$
 7:         Set $j = 1$ & $\alpha_{old} = \alpha_{tr}$            ▷ Initialize parameters
 8:         Compute $\hat{w}_i^s$ using Eq. (3.20)
 9:         while  do                                            ▷ Initiate "virtual" loop
10:             Resample $N$ samples: $\boldsymbol{\theta}_i^s \sim \hat{w}_i^s$
11:             Set $\boldsymbol{\theta}_i^s = \boldsymbol{\theta}_{i,1}$ in ensemble $\vec{\boldsymbol{\theta}}_1$     ▷ Initiate ensemble
12:             Update $\vec{\boldsymbol{\theta}}_1$ with 1 iteration of AIES (see Algorithm 6)     ▷ MCMC step
13:             Compute $\alpha$ using Eq. (4.10)
14:             Compute $u_{nom}$ using Eq. (4.16)               ▷ Tuning the step-size
15:             Set $u^{s+1} = \max(u_{nom}, 1.01)$
16:             if $|\alpha - \alpha_{old}| < 0.1 \cdot \alpha_{tr}$ then     ▷ Check termination criteria
17:                 Break                                        ▷ Exit "virtual" loop
18:             end if
19:             Set $\alpha_{old} = \alpha$
20:             Set $j = j + 1$
21:         end while                                            ▷ End "virtual" loop
22:         Set updated ensemble $\vec{\boldsymbol{\theta}}_1$ as $\boldsymbol{\theta}_i^s \sim P(\boldsymbol{\theta}^s|\boldsymbol{D}^{1:s}, M)$
23:         Compute $\boldsymbol{\theta}_i^{s+1}$ using $T(\boldsymbol{\theta}^{s+1}|\boldsymbol{\theta}^s)$     ▷ Set as new prior samples
24:         Compute PDF of $P(\boldsymbol{\theta}_i^{s+1}|D^{1:s})$ using Eq. (7.3)     ▷ Set as new prior PDF
25:         Compute $P(\boldsymbol{D}^{1:s}|M)$ using Eq. (4.1)
26:     end while
27: end procedure
```

7.5 Chapter conclusion

The chapter has introduced the concept of the Sequential Ensemble Monte Carlo sampler and how it can be implemented to infer the time-varying parameters that are considered constant while a sequence of observations is obtained but may vary between the different sequences of such observations. The proposed sampler seeks to be a robust and efficient variant of the Sequential Monte Carlo sampler. The proposed sampler includes 2 new features: 1) the implementation of the affine-invariant ensemble sampler in place of the traditional Metropolis-Hasting sampler in the MCMC step; and 2) the introduction of an adaptive tuning algorithm making the approach "tune-free" for users. For the second feature, to ensure robustness in converging the sampler's acceptance rate to the target value upon the first iteration, additional "virtual iterations" in the form of while loops are introduced in the first iteration which terminates upon satisfying the defined termination criteria. This serves to tune the step-size parameter of the sampler in an iterative manner which, in turn, controls the acceptance rate.

In addition, a novel analysis on the acceptance rates of the proposed sampler is being studied which indicated the existence of inherent bounds on the acceptance rate values. Through such findings, a qualitative explanation is provided behind the presence of an inherent minimum value and maximum value on the acceptance rate. This also provides a rationale as to why the analyst need not be concerned with the acceptance rate of the sampler falling below the optimal value of 0.15 and that it is only essential to ensure that the acceptance rate value is not too high (i.e. between 0.8 and 0.9).

Finally, it needs to be highlighted that a significant drawback of the sampler is the relative complexity in coding the Sequential Ensemble Monte Carlo algorithm from scratch compared to the Sequential Monte Carlo sampler with Metropolis-Hastings kernel. To address this issue, access to the MATLAB code is provided on GitHub: `https://github.com/Adolphus8/Sequential_Ensemble_Monte_Carlo.git`

Following the chapter, the implementation of the proposed Sequential Ensemble Monte Carlo sampler will be demonstrated on an experimental example involving experimental data and a physical structure to which details and results are presented in Chapter 8 of the dissertation. Through this example, a comparison in the sampling performance and the resulting estimates on the inferred parameters (both time-varying and static) will be made between the proposed Sequential Ensemble Monte Carlo sampler and the Sequential Monte Carlo sampler with Metropolis-Hastings kernel.

Chapter 8

On-line Identification of Time-varying Model parameters for Structural Health Monitoring

Abstract

The chapter illustrates the implementation of the proposed Sequential Ensemble Monte Carlo sampler, presented in Chapter 7, in an experimental example involving a $SDoF$ single-storey Shear Frame structure subjected to a Coulomb friction. The study consists of 2 parts: In the first part, the proposed sampler is implemented alongside the traditional Sequential Monte Carlo sampler in identifying the most probable Markov kernel to describe the evolution of the Coulomb friction with time. In the second part of the study, the time-varying Coulomb friction as well as the time-invariant natural frequency of the structure and measurement errors are inferred. As such, there are 4 objectives to this study: 1) to validate the proposed sampler in the Bayesian identification of the aforementioned parameters using actual experimental data; 2) to compare and verify the inference results obtained by the proposed sampler against that of the Sequential Monte Carlo sampler; 3) to demonstrate the robustness of the proposed sampler in controlling the acceptance rates under challenging settings; and 4) to assess the robustness of the proposed sampler in identifying the most probable Markov kernel under uncertainty.

8.1 Background

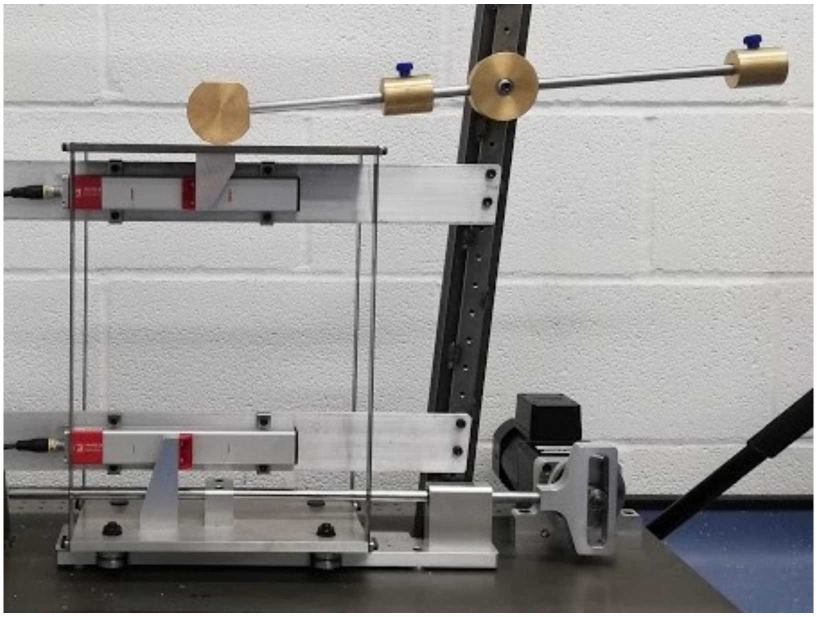


Figure 8.1: Physical set-up of the *SDoF* single-storey shear frame structure subjected to Coulomb friction.

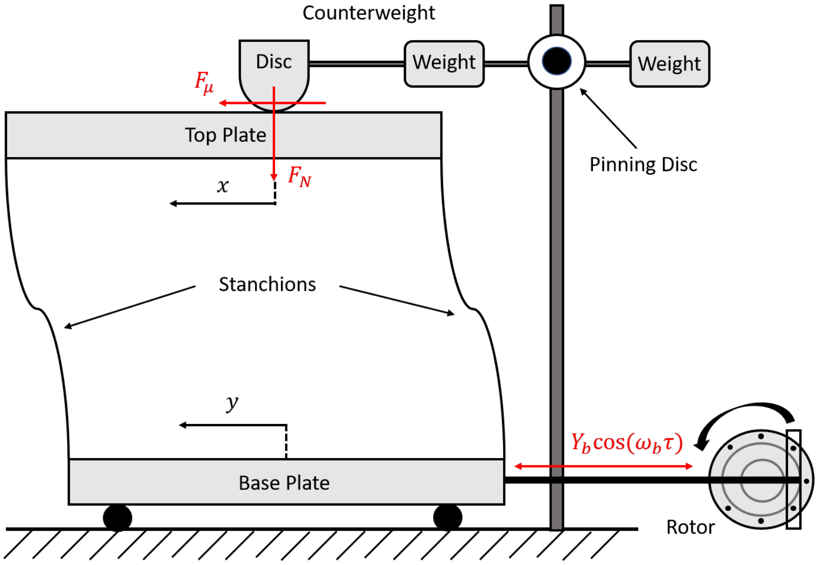


Figure 8.2: Schematic diagram of the *SDoF* single-storey shear frame structure subjected to Coulomb friction. Image adapted from [9].

In this experimental example, the effect of the time-varying friction force on the response dynamics of a physical structure will be investigated. Such application example is specifically chosen due to its importance in assessing the dynamic performance of structures and avoiding friction-related failures [9, 315]. This investigation is conducted using a single-storey shear frame with a Coulomb friction contact, subjected to a harmonic base-excitation. The physical set-up and schematic diagram of the structure are presented in Figures 8.1 and 8.2 respectively. Detailed description to the physical set-up can be found in [9].

The objective of this investigation are the following: 1) to evaluate the capability of the proposed SEMC sampler in its ability to infer the values of Coulomb friction force at each time step sequence, the time-invariant natural frequency of the structure, and the measurement error using sets of actual experimental data obtained sequentially; 2) to verify the inference results obtained by the proposed SEMC sampler against the SMC-MH sampler; and 3) to evaluate the model identification capability of the SEMC sampler will also be investigated in identifying the most probable Markov kernel to model the time-varying characteristic of the Coulomb friction force under uncertainty.

8.2 Physics-based model of the structure

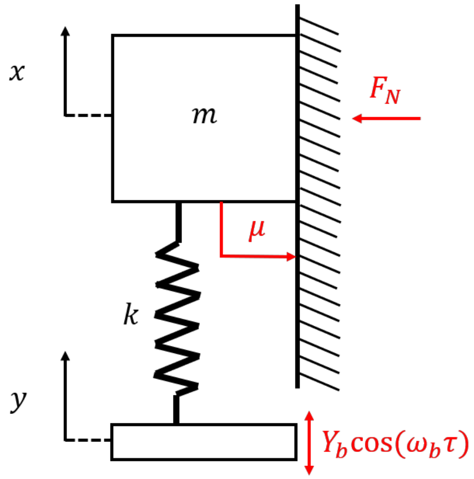


Figure 8.3: Spring-mass representation of the $SDoF$ single-storey shear frame structure subjected to Coulomb friction.

The building can be modelled as Spring-Mass-Damped system as shown in Figure 8.3 where the mass m and the stiffness k represent the participating mass and stiffness of the first vibrating mode of the structure [9, 315]. The Coulomb friction force F_μ is generated as a result of contact between the mass m and a fixed wall and is obtained as the product of a friction coefficient μ and the normal contact force F_N.

To study the response dynamics of the structure as well as the parameters to be inferred in this problem, it is possible to write the governing equation of this $SDoF$ model in a dimensionless form as [315]:

$$r^2 \cdot \frac{d^2\tilde{x}}{d{\tau_b}^2} + \tilde{x} + \beta(t) \cdot \operatorname{sgn}\left(\frac{d\tilde{x}}{d\tau_b}\right) = \cos\left(\tau_b\right) \tag{8.1}$$

where $\tilde{x}$ is the dimensionless response of m, r is the frequency ratio, $\beta(t)$ is the time-varying friction ratio, and τ_b is the dimensionless time parameter. The dimensionless frequency ratio r is defined as [315]:

$$r = \frac{\omega_b}{\omega_n} \tag{8.2}$$

where $\omega_n = \sqrt{\frac{k}{m}}$ is the natural frequency of the structure which was determined experimentally [9] to be 19.572 rad/s and ω_b is the driving frequency of the harmonic base-excitation. Hence, the dimensionless time τ_b in Eq. (8.1) is defined as: $\tau_b = \omega_b \cdot \tau$, where τ is the physical time parameter. The dimensionless force ratio $\beta(t)$ is defined as [315]:

$$\beta(t) = \frac{F_\mu(t)}{kY_b} \tag{8.3}$$

where Y_b is the driving displacement amplitude by the rotor. Thus, kY_b is the driving force amplitude whose value was measured experimentally to be 2.50 N. Hence, the dimensionless response $\tilde{x}$ is defined as [315]:

$$\tilde{x} = \frac{x}{Y_b} \tag{8.4}$$

where x is the response displacement of m. According to Den-Hartog's theory [316], under the assumptions of continuous and symmetric response, the steady-state solution of Eq. (8.1) can be obtained analytically as [315]:

$$\tilde{x}_{num}(\tau_b) = \begin{cases} \tilde{x}(\tau_b) & \text{for } \tau_b \in [0, \pi) \\ -\tilde{x}(\tau_b - \pi) & \text{for } \tau_b \in [\pi, 2\pi) \end{cases} \tag{8.5}$$

where $\tilde{x}(\tau_b)$ can be evaluated as:

$$\tilde{x}(\tau_b) = \tilde{x}_0 \cdot \cos(\tau_b) + \beta(t)U \cdot \sin(\tau_b) + \beta(t) \cdot \left[1 - \cos\left(\frac{\tau_b}{r}\right) - Ur \cdot \sin\left(\frac{\tau_b}{r}\right)\right] \tag{8.6}$$

In Eq. (8.6), the damping function U is defined as:

$$U = \frac{\sin(\pi/r)}{r \cdot [1 + \cos(\pi/r)]} \tag{8.7}$$

while the dimensionless response amplitude $\tilde{x}_0$ can be evaluated as:

$$\tilde{x}_0 = \sqrt{\left(\frac{1}{1 - r^2}\right)^2 - (\beta(t) \cdot U)^2} \tag{8.8}$$

Details on the derivation of the above terms can be found in [315].

8.3 Data collection

For this study, 4 different values of Coulomb friction force $F_\mu(t)$ are considered to simulate its time-varying aspect for $t = \{1, 2, 3, 4\}$ months. This variation in $F_\mu(t)$ can be simulated by varying the configuration of the weights in the counterweight system seen in Figure 8.2. These values of $F_\mu(t)$ are: $\{1.435, 0.980, 0.662, 0.217\}$ N respectively. For each $F_\mu(t)$, 9 sets of phase angle data ϕ are collected across 9 chosen values of frequency ratio r. The phase angle ϕ is chosen as the response data due to its high-degree of sensitivity to the variation in $F_\mu(t)$ as shown in [9]. The reference values for the 9 chosen values of r, which can be obtained by adjusting ω_b (i.e. see Eq. (8.2)), are: $r_{nom} = \{0.65, 0.80, 0.95, 1.10, 1.25, 1.40, 1.55, 1.70, 1.85\}$. The experimental procedure to obtain the phase angles ϕ from given values of $F_\mu(t)$ and r can be found in [9].

The experimental measurements of ϕ, r, and $F_\mu(t)$ are obtained by Dr. Luca Marino and are shown in Table 8.1 and in Figure 8.4. In Figure 8.4, the Den-Hartog's boundary denotes the boundary between the continuous motion and the stick-slip regime for the dynamic response of the top plate under Coulomb friction, while the continuous colored lines represent the true model output of ϕ, given $F_\mu(t_s)$, for the different values of r within the continuous motion regime. In addition, the values of the driving frequency ω_b measured from each test are presented in Table 8.2.

Based on the experimentally-obtained values of $F_\mu(t)$, 2 choices of Markov kernels are identified to model the time-based degradation of $F_\mu(t)$:

Exp.	r_{nom}	$F_\mu(t_1) = 1.435\ N$		$F_\mu(t_2) = 0.980\ N$		$F_\mu(t_3) = 0.662\ N$		$F_\mu(t_4) = 0.217\ N$	
		r	ϕ $[deg]$	r	ϕ $[deg]$	r	ϕ $[deg]$	r	ϕ $[deg]$
1	0.65	0.649	46.464	0.637	27.217	0.624	16.622	0.652	6.496
2	0.80	0.791	41.492	0.789	26.269	0.807	16.900	0.796	4.292
3	0.95	0.952	41.864	0.944	29.936	0.941	14.934	0.936	5.839
4	1.10	1.098	132.661	1.099	147.498	1.123	162.318	1.110	174.223
5	1.25	1.278	137.022	1.253	150.104	1.255	160.757	1.262	174.832
6	1.40	1.407	129.795	1.406	152.246	1.409	156.074	1.392	171.955
7	1.55	1.557	136.944	1.549	152.011	1.540	161.960	1.548	173.666
8	1.70	1.706	131.314	1.694	152.008	1.711	157.884	1.715	171.698
9	1.85	1.848	134.294	1.849	153.251	1.833	161.017	1.860	169.833

Table 8.1: Experimental results of r and ϕ obtained for the respective $F_\mu(t_s)$.

Exp.	$F_\mu(t_1) = 1.435\ N$	$F_\mu(t_2) = 0.980\ N$	$F_\mu(t_3) = 0.662\ N$	$F_\mu(t_4) = 0.217\ N$
	ω_b $[rad/s]$	ω_b $[rad/s]$	ω_b $[rad/s]$	ω_b $[rad/s]$
1	12.696	12.462	12.213	12.751
2	15.487	15.444	15.791	15.576
3	18.639	18.478	18.415	18.318
4	21.480	21.500	21.983	21.729
5	25.015	24.514	24.569	24.694
6	27.540	27.522	27.585	27.239
7	30.468	30.321	30.141	30.296
8	33.390	33.149	33.478	33.558
9	36.165	36.195	35.880	36.398

Table 8.2: Experimental values of ω_b used for the respective $F_\mu(t_s)$.

$$T_1 : \quad F_\mu(t_{s+1}) = F_\mu(t_s) - 0.375 + \nu_1 \tag{8.9}$$

$$T_2 : \quad F_\mu(t_{s+1}) = \exp\left[-0.470\right] \cdot F_\mu(t_s) + \nu_2 \tag{8.10}$$

whereby $s = 1, \ldots, 4$ denotes the time sequence index, and ν_1 and ν_2 are the zero-mean Normally-distributed process "noise" terms with the respective standard deviations: $\{\sigma_1, \sigma_2\} = \{0.040, 0.090\}\ N$. The parameters of the Markov kernels are obtained using a curve-fitting procedure via the Least-squares method on the experimentally-obtained values of $F_\mu(t)$. The corresponding nominal models Γ_1 and Γ_2 (i.e. the Kernel models without the noise term as defined in Eq. (7.1)) are shown in Figure 8.5. It needs to be added that while in this example the parameters of the Markov kernel are assumed to be known, this is not always the case

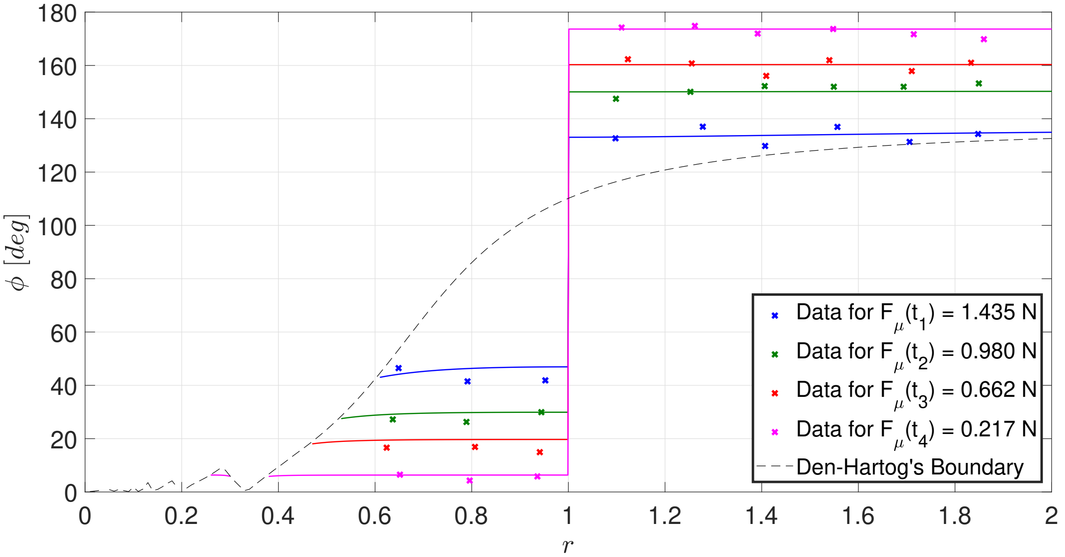


Figure 8.4: Plots of r and ϕ for the corresponding values of $F_\mu(t_s)$ for $s = \{1, \ldots, 4\}$.

in general. In such cases, the parameters of the Markov kernel can also be included in the set of inferred parameters through Bayesian inference [37].

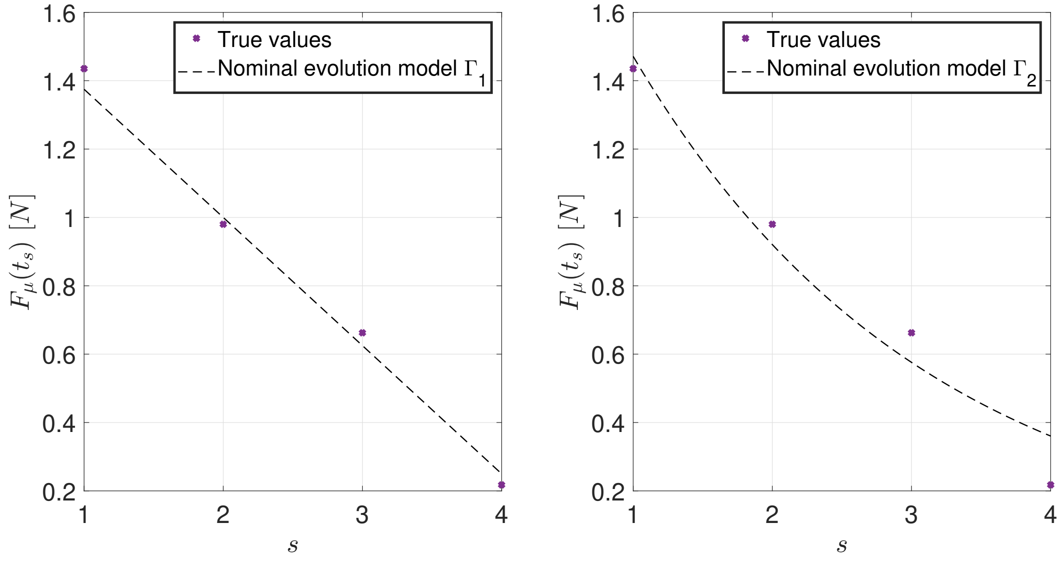


Figure 8.5: Scatterplot of the true values of $F_\mu(t_s)$ across the time step sequences s with the nominal evolution models Γ_1 and Γ_2.

8.4 Sequential Bayesian inference set-up

The sequential Bayesian inference procedure is done for $\boldsymbol{\theta}^s = (F_\mu(t_s), \omega_n, \sigma_\phi, \sigma_r)$, where σ_ϕ and σ_r are the standard deviations of the respective "noise" associated with the experimentally-obtained values of ϕ and r. The parameters ω_n, σ_ϕ and σ_r are assumed to be time-invariant and it needs to be noted that σ_ϕ and σ_r are are internal parameters of the likelihood function and are not used in the models to predict $F_\mu(t_s)$ and ω_n. This gives rise to a 4-dimensional Bayesian inference problem for the estimation of the aforementioned parameters at each time sequence t_s.

The initial priors at $s = 1$ for each of the inferred parameters are set as Uniform priors whose bounds are listed in Table 8.3. For $s \geq 2$ the prior is the predictive distribution that is derived using Eq. (7.3) by propagating the samples obtained at previous time step $s - 1$ through the Markov kernel. The likelihood function for each time sequence t_s is set to follow a Normal distribution. Assuming independence between individual observations of ϕ and r as well as between data-sets obtained at different time step sequences s, the likelihood function is defined as:

$$P(\boldsymbol{D}^s | \boldsymbol{\theta}^s, \hat{\phi}, \hat{r}) = \prod_{q=1}^{9} \frac{1}{2\pi \cdot \sigma_r \cdot \sigma_\phi} \cdot \\ \exp\left[-\frac{(r_{nom}^q - \hat{r}(\omega_b^{q,s}, \omega_n))^2}{2 \cdot {\sigma_r}^2} - \frac{\left(\phi^{q,s} - \hat{\phi}(r^{q,s}, F_\mu(t_s))\right)^2}{2 \cdot {\sigma_\phi}^2} \right] \quad (8.11)$$

where r_{nom}^q, ω_b^q and ϕ^q denote respectively the q^{th} value/observation of r_{nom}, ω_b, and ϕ obtained at time sequence t_s for $q = 1, \ldots, 9$, $\hat{r}$ denotes the model used to compute r, $\hat{\phi}$ is the model used to compute ϕ, and $\boldsymbol{D}^s = (\phi, r)^s$ denotes the data set obtained at time t_s.

Parameter	Bounds	Units
$F_\mu(t_s)$	$[0.01, 100]$	N
ω_n	$[0.01, 100]$	rad/s
σ_ϕ	$[0.001, 10]$	deg
σ_r	$[0.001, 1]$	$-$

Table 8.3: The bounds assigned to the Uniform prior for the respective inferred parameters.

Details on models $\hat{r}$ and $\hat{\phi}$ are as such. The model $\hat{r}$ evaluates the analytical solution for

r from a given value of ω_b and ω_n according to Eq. (8.2). The computation procedure by the model $\hat{\phi}$ to evaluate the analytical solution of ϕ from a given value of $F_\mu(t_s)$ and r consists of 3 main steps [316]:

In the first step, the algorithm computes $\beta(t_s)$ with the input value of $F_\mu(t_s)$ using Eq. (8.3). In order to verify the assumption of continuous response, the value of β corresponding to the boundary between continuous and stick-slip regimes, shown in Figure 8.4, is also computed as [316]:

$$\beta_{lim} = \sqrt{\frac{1}{\left(U^2 + \frac{1}{r^4}\right) \cdot (1 - r^2)^2}} \tag{8.12}$$

If $\beta(t_s) > \beta_{lim}$, the condition for a continuous motion is not satisfied. Therefore, the algorithm proceeds to assign a NaN (i.e. Not a Number) value for ϕ and the procedure terminates here. Otherwise, the algorithm proceeds to the next step. In the second step, the analytical steady-state response solution $\tilde{x}_{num}(\tau_b)$ is computed for $\tau_b \in [0, 2\pi)$ from Eq. (8.5) and the numerical excitation function $\tilde{y}_{num}(\tau_b)$, expressed as [316]:

$$\tilde{y}(\tau_b) = \cos(\tau_b + \varphi) \tag{8.13}$$

where:

$$\varphi = \text{atan2}\left[-\beta(t_s) \cdot U \cdot (1 - r^2),\ \tilde{x}_0 \cdot (1 - r^2)\right] \tag{8.14}$$

is also computed for $\tau_b \in [0, 2\pi)$. Finally, in the last step, the algorithm proceeds to compute the phase angle ϕ between the excitation and the response functions. This is done by obtaining their respective dimensionless frequency spectra $\tilde{x}_{FFT}(\tilde{f})$ and $\tilde{y}_{FFT}(\tilde{f})$ using the FFT algorithm [268, 317]. The dimensionless frequency is here defined as $\tilde{f} = \frac{2\pi \cdot f}{\omega_b}$, where f is the frequency variable in the FFT-space. From there, the phase angle is computed at $\tilde{f} = 1$ (i.e. resonance) following [9]:

$$\phi = \arg\{\tilde{x}_{FFT}(\tilde{f} = 1)\} - \arg\{\tilde{y}_{FFT}(\tilde{f} = 1)\} \tag{8.15}$$

A pseudo-algorithm describing the above procedure is presented in Algorithm 10. It needs to be acknowledged that the MATLAB codes to the model $\hat{\phi}$ used in the Bayesian inference procedure is provided by Dr. Luca Marino. In the event $\hat{\phi} = \text{NaN}$, the likelihood function $P(\boldsymbol{D}^s | \boldsymbol{\theta}^s, \hat{\phi}, \hat{r})$ returns a 0.

Algorithm 10 Pseudo-algorithm of model $\hat{\phi}$

```
procedure (Compute $\phi$ from $F_{\mu}(t_s)$ and $r$)
    Compute $\beta(t_s)$ with $F_{\mu}(t_s)$ using Eq. (8.3)
    Compute $\beta_{lim}$ with $r$ using Eq. (8.12)
    if $\beta > \beta_{lim}$ then                ▷ Steady, continuous motion condition not satisfied
        Set $\phi$ = NaN
    else
        Compute $\tilde{x}_{num}(\tau_b)$ using Eq. (8.5)
        Compute $\tilde{y}_{num}(\tau_b)$ using Eq. (8.13)
        Execute FFT on $\tilde{x}_{num}(\tau_b)$ to generate $\tilde{x}_{FFT}(\tilde{f})$
        Execute FFT on $\tilde{y}_{num}(\tau_b)$ to generate $\tilde{y}_{FFT}(\tilde{f})$
        Set $\tilde{f} = 1$
        Compute $\phi$ using Eq. (8.15)
    end if
end procedure
```

8.4.1 Results and discussions

From the posteriors $P(\boldsymbol{\theta}^s|\boldsymbol{D}^{1:s}, M)$, 1000 samples are generated at each given time step sequence s by the proposed SEMC and SMC-MH samplers for each given Markov kernel. For the case of the Markov kernel T_1, the total sampling time by the SEMC and SMC-MH samplers are 98.70 s and 111.25 s respectively. For the case of the Markov kernel T_2, the total sampling time by the SEMC and SMC-MH samplers are 95.40 s and 114.39 s respectively.

As a metric to quantify how probable a given Markov kernel is in modelling the variation of $F_{\mu}(t_s)$ across the time step sequences s, the log-evidence term $\log\left[P(\boldsymbol{D}^{1:s}|M)\right]$ is used and computed at each s for each Markov kernel by both the SEMC and SMC-MH samplers. The numerical results are summarised in Tables 8.4 and 8.5 while a graphical plot illustration is provided in Figure 8.6. As seen in Tables 8.4 and 8.5, $\log\left[P(\boldsymbol{D}^{1:s}|M)\right]$ is consistently higher for the case of T_1 which indicates that T_1 is the most probable Markov kernel to represent the variation of $F_{\mu}(t_s)$ across s.

As a metric to quantify and compare the sampling performance between the SEMC and SMC-MH samplers, the effective sample size N^s_{eff} is used. The metric N^s_{eff} is computed at each time step sequence s following [318]:

$$N_{eff} = \frac{1}{\sum_{i=1}^{N}(\hat{w}^s_i)^2} \tag{8.16}$$

where $\hat{w}^s_i$ is computed using Eq. (3.20). Should the value of N^s_{eff} fall below a threshold value

s	$\log\left[P(\boldsymbol{D}^{1:s}\|M)\right] \mid T_1$	$\log\left[P(\boldsymbol{D}^{1:s}\|M)\right] \mid T_2$	Difference
1	-25.066	-28.458	3.392
2	-38.014	-53.547	15.533
3	-45.478	-74.898	29.420
4	-49.206	-93.903	44.697

Table 8.4: Log-evidence computed for the model identification procedure using the SEMC sampler.

s	$\log\left[P(\boldsymbol{D}^{1:s}\|M)\right] \mid T_1$	$\log\left[P(\boldsymbol{D}^{1:s}\|M)\right] \mid T_2$	Difference
1	-28.824	-29.963	1.139
2	-36.405	-53.949	17.544
3	-42.136	-74.300	32.164
4	-52.174	-93.976	41.802

Table 8.5: Log-evidence computed for the model identification procedure using the SMC-MH sampler.

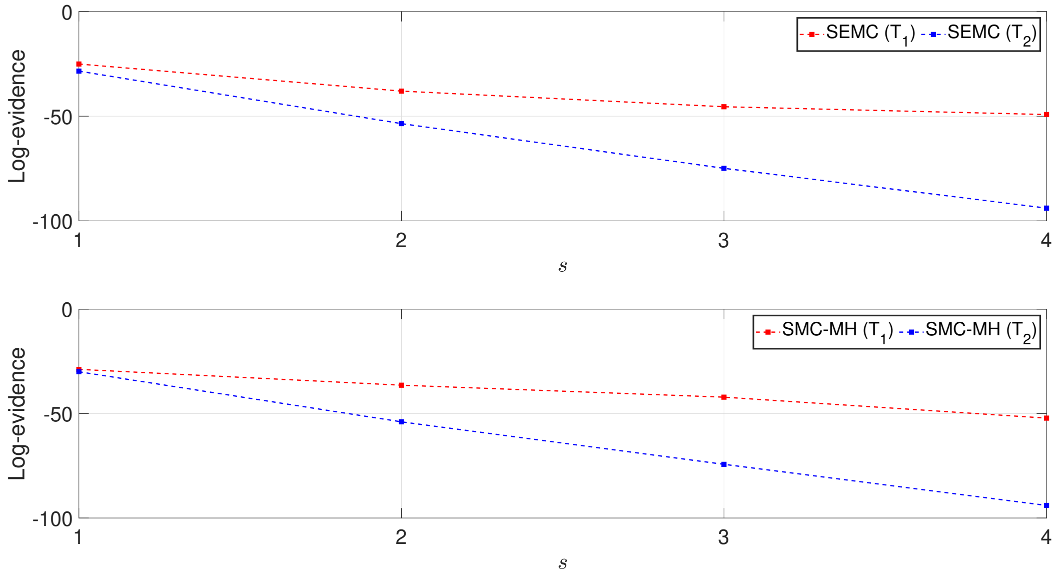


Figure 8.6: Graphical plot illustration of the results for log-evidence computed across the time step sequences s for each Markov kernel by the respective samplers.

of $\frac{N}{2}$ (i.e. 500 in this study) [52, 319], it indicates the presence of degeneracy where most of the samples have weights $\hat{w}_i^s$ that are close to 0 (i.e. see Section 3.2.1 for an illustration). As highlighted in [52], the choice of forward kernels (i.e. the choice of MCMC move kernels) or backward kernels can have a dramatic effect on the N^s_{eff}. The resulting statistics of N^s_{eff} for both the SEMC and SMC-MH samplers given both Markov kernels are presented in Figure 8.7 and Table 8.6. From the figure, it can be seen that the values of N^s_{eff} across the time step sequences s is generally higher for the case of the SEMC sampler compared to the SMC-MH sampler for both Markov kernels. This is supported from the numerical results in Table 8.6. In addition, it can also be observed that the value of N^s_{eff} falls below the threshold value of 500 only once at time step sequence $s = 1$ in the case of the SEMC sampler for both Markov kernels. In the case of the SMC-MH sampler, the value of N^s_{eff} falls below the threshold value at time step sequence $s = 1$ for the case of Markov kernel T_1 and twice at time step sequence $s = 1$ and $s = 2$ for the case of Markov kernel T_2. These results indicate a better mixing performance as a result of the implementation of the AIES as the MCMC move kernel of the SEMC sampler compared to the MH in the case of the SMC-MH sampler.

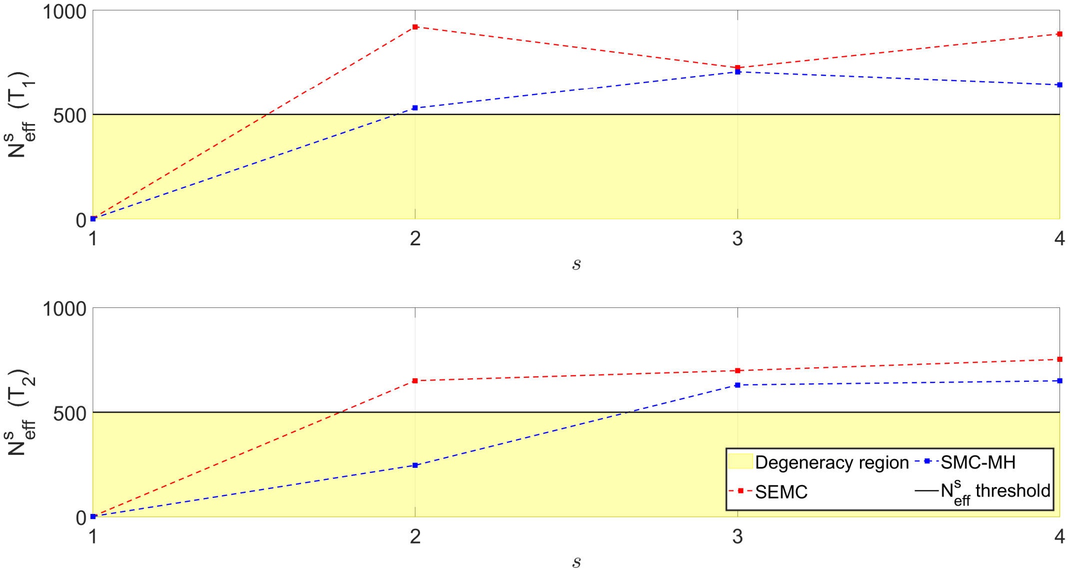


Figure 8.7: Graphical plots summarising the statistics of the effective sample size across the time step sequences s for each Markov kernel by the respective samplers .

The results of acceptance rates are illustrated in Figure 4.10. From the figure, it can be observed that a superior convergence of the acceptance rates is achieved by the SEMC sampler for both the Markov kernels T_1 and T_2 right from time step sequence $s = 1$ while the SMC-MH sampler reached convergence from $s = 3$ onwards. This demonstrates the effectiveness of the adaptive tuning algorithm within the SEMC sampler in ensuring the convergence of the acceptance rates.

s	SEMC		SMC-MH	
	T_1	T_2	T_1	T_2
1	2.637	1.800	1.001	1.755
2	921.041	650.795	530.409	246.153
3	725.305	698.942	705.795	630.369
4	887.287	752.461	643.819	650.163

Table 8.6: Statistics of the effective sample size by the SEMC and SMC-MH samplers given each Markov kernel. The threshold effective sample size is 500.

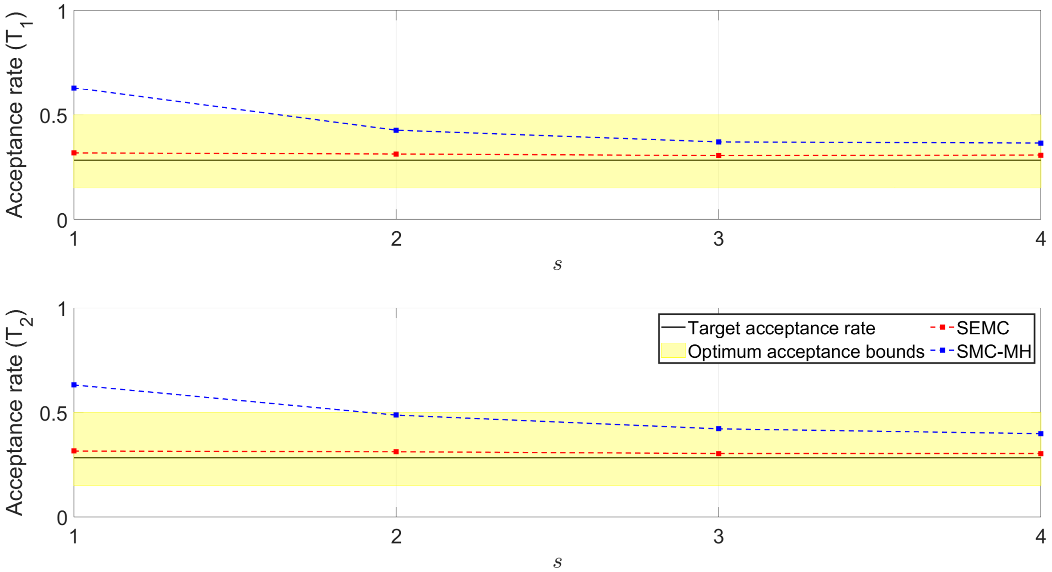


Figure 8.8: Graphical plots summarising the statistics of the acceptance rates across the time step sequences s by the SEMC and SMC-MH samplers for each Markov kernel. Target acceptance rate: 0.283 (i.e. see Eq. (4.2)).

In reporting the statistics of each of the inferred parameters, the sample mean estimates, the Monte Carlo standard error σ_{SE}, and the corresponding 3-σ_{SE} bounds are presented. The Monte Carlo standard error σ_{SE} is computed from Eq. (3.4) with batch size $n_b = 100$.

The resulting statistics of the estimates for $F_\mu(t_s)$ across each time step sequence s are obtained for each choice of the Markov kernel and summarised in Figure 8.9 and in Tables 8.7 to 8.8. Figure 8.9 shows the identified value of the parameter $F_\mu(t_s)$ with the corresponding error bars reflecting the 3-σ_{SE} bounds. For both the SEMC and SMC-MH samplers, their estimates of $F_\mu(t_s)$ generally follow the trend defined by the evolution model which is defined within the Markov kernel (i.e. see Eq. (8.9) and Eq. (8.10)). The SEMC estimates for $F_\mu(t_s)$

given T_1 are generally closer to the true values compared to the estimates using T_2 while the standard error of the estimates at each time step sequence s are generally similar between the different Markov kernels. A similar trend is also observed for the SMC-MH estimates. This is due to the fact that T_1 describes better the change of $F_\mu(t_s)$ across the time step sequences s compared to T_2. Hence, this set of results illustrates the direct influence of the choice of Markov kernel on not just the trajectory of the time-varying estimates of $F_\mu(t_s)$, but also the accuracy of the estimates of $F_\mu(t_s)$ across any given s.

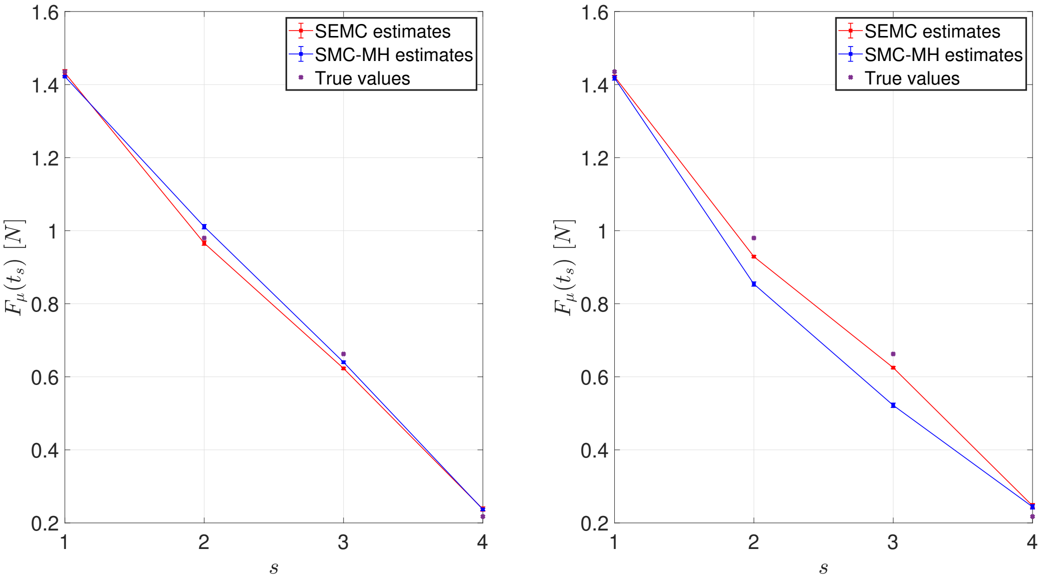


Figure 8.9: Graphical plots summarising the statistics of the parameter estimates of $F_\mu(t_s)$, along with the 3-σ_{SE} bounds, by the SEMC and SMC-MH samplers given the Markov kernels T_1 (left) and T_2 (right).

s	True value	SEMC | T_1			SEMC | T_2		
		$E[F_\mu(t_s)]$	σ_{SE}	3-σ_{SE} bounds	$E[F_\mu(t_s)]$	σ_{SE}	3-σ_{SE} bounds
	$[N]$	$[N]$	$[N]$	$[N]$	$[N]$	$[N]$	$[N]$
1	1.435	1.432	0.003	$[1.423, 1.441]$	1.422	0.001	$[1.419, 1.425]$
2	0.980	0.966	0.002	$[0.960, 0.972]$	0.929	0.001	$[0.926, 0.932]$
3	0.662	0.623	0.001	$[0.620, 0.626]$	0.625	0.001	$[0.622, 0.628]$
4	0.217	0.238	0.002	$[0.232, 0.244]$	0.247	0.002	$[0.241, 0.253]$

Table 8.7: Statistics of the parameter estimates of $F_\mu(t_s)$ by the SEMC sampler given each Markov kernel.

s	True value	SMC-MH | T_1			SMC-MH | T_2		
		$E[F_\mu(t_s)]$	σ_{SE}	3-σ_{SE} bounds	$E[F_\mu(t_s)]$	σ_{SE}	3-σ_{SE} bounds
	$[N]$	$[N]$	$[N]$	$[N]$	$[N]$	$[N]$	$[N]$
1	1.435	1.422	0.001	$[1.419, 1.425]$	1.418	0.002	$[1.412, 1.424]$
2	0.980	1.011	0.002	$[1.005, 1.017]$	0.854	0.002	$[0.848, 0.860]$
3	0.662	0.640	0.001	$[0.637, 0.643]$	0.522	0.002	$[0.516, 0.528]$
4	0.217	0.237	0.001	$[0.234, 0.240]$	0.244	0.002	$[0.238, 0.250]$

Table 8.8: Statistics of the parameter estimates of $F_\mu(t_s)$ by the SMC-MH sampler given each Markov kernel.

The resulting statistics of the estimates for the parameters, ω_n, σ_ϕ, and σ_r, across the time step sequence s, obtained for each choice of the Markov kernel, are shown in Figures 8.10 and 8.11 while the corresponding numerical results summarised in Tables 8.9 to 8.14. Like in Figure 8.9, the error bars correspond to the 3-σ_{SE} bounds. For the case of ω_n, the SEMC and SMC-MH estimates given either choice of the Markov kernel both converge to the experimentally identified values (see Section 8.2). This indicates the effectiveness of the samplers in inferring ω_n. However, it can be observed from Tables 8.9 to 8.14 that the Monte Carlo standard error of the estimates is generally smaller for the case of the SEMC sampler compared to the SMC-MH sampler for both Markov kernels. This indicates that the SEMC sampler is able to provide more precise sample estimates on ω_n compared to the SMC-MH sampler across the time step iterations s.

The reference values for σ_ϕ and σ_r, denoted as σ_ϕ^{ref} and σ_r^{ref} respectively, are set as the mean of the Root Mean-Squared-Error (RMSE) across t_s which are computed as follows:

$$\sigma_\phi^{ref} = \frac{1}{4}\sum_{s=1}^{4}\sqrt{\frac{1}{9}\sum_{q=1}^{9}\left[\phi^{q,s} - \hat{\phi}\left(F_\mu(t_s),\ r^{q,s}\right)\right]^2} \tag{8.17}$$

$$\sigma_r^{ref} = \frac{1}{4}\sum_{s=1}^{4}\sqrt{\frac{1}{9}\sum_{q=1}^{9}\left(r^{q,s} - r_{nom}\right)^2} \tag{8.18}$$

The SEMC sampler provides an estimate of σ_ϕ close to the reference values given either choice of the Markov kernel. However, the 3-σ_{SE} bounds of the estimates given the Markov kernel T_2 is significantly larger than that for Markov kernel T_1 (i.e. see Tables 8.11 and 8.12). A similar observation is made for the SMC-MH estimates of σ_ϕ although its estimates are significantly larger than that by the SEMC sampler. In estimating σ_r, neither the results obtained by the SEMC and SMC-MH samplers come close to the reference value of $\sigma_r^{ref} = 0.010$ although the sample estimates obtained using the SEMC sampler yields values which are closer to σ_r^{ref} given either choice of the Markov kernel. In general, the Monte Carlo standard error of the estimates by the SEMC sampler for σ_ϕ and σ_r are generally smaller than that by the SMC-MH sampler for the respective Markov kernel. This indicates a better precision of the sample estimates by the SEMC sampler compared to the SMC-MH sampler.

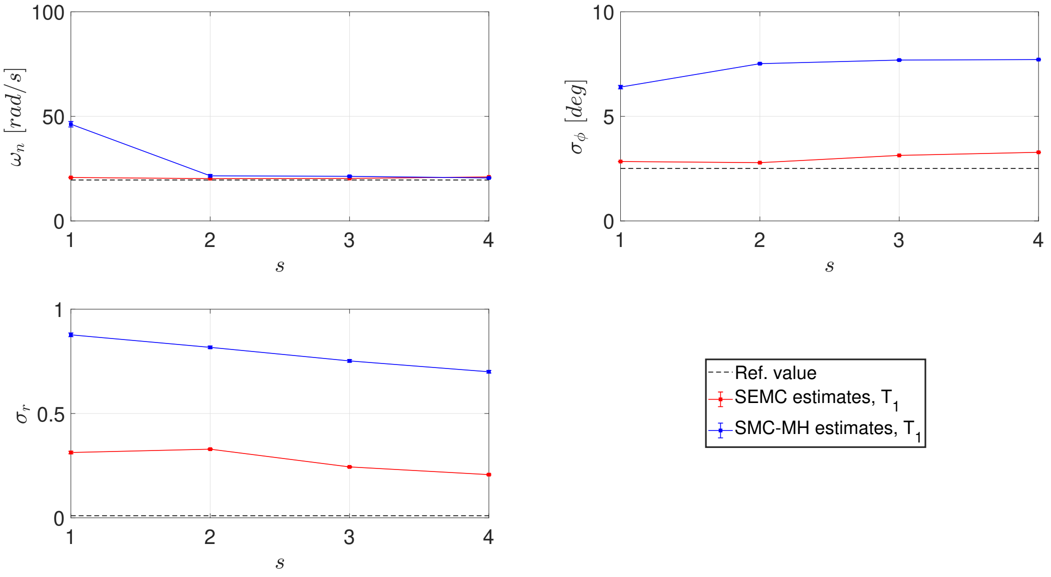


Figure 8.10: Graphical plots summarising the statistics of the parameter estimates of ω_n, σ_ϕ, and σ_r, along with the 3-σ_{SE} bounds, by the SEMC and SMC-MH samplers given the Markov kernel T_1. The reference values for the respective parameters are: $\{\omega_n, \sigma_\phi, \sigma_r\} = \{19.572\,\text{rad/s}, 2.512^o, 0.010\}$.

s	Experimentally identified value	SEMC | T_1			SEMC | T_2		
		$E[\omega_n]$	σ_{SE}	3-σ_{SE} bounds	$E[\omega_n]$	σ_{SE}	3-σ_{SE} bounds
	$[rad/s]$	$[rad/s]$	$[rad/s]$	$[rad/s]$	$[rad/s]$	$[rad/s]$	$[rad/s]$
1	19.572	20.766	0.050	$[20.616, 20.916]$	21.014	0.095	$[20.729, 21.299]$
2	19.572	20.224	0.012	$[20.188, 20.260]$	21.622	0.106	$[21.304, 21.940]$
3	19.572	20.329	0.014	$[20.287, 20.371]$	20.692	0.049	$[20.545, 20.839]$
4	19.572	21.043	0.011	$[21.010, 21.076]$	20.462	0.029	$[20.375, 20.549]$

Table 8.9: Statistics of the parameter estimates of ω_n by the SEMC sampler given each Markov kernel.

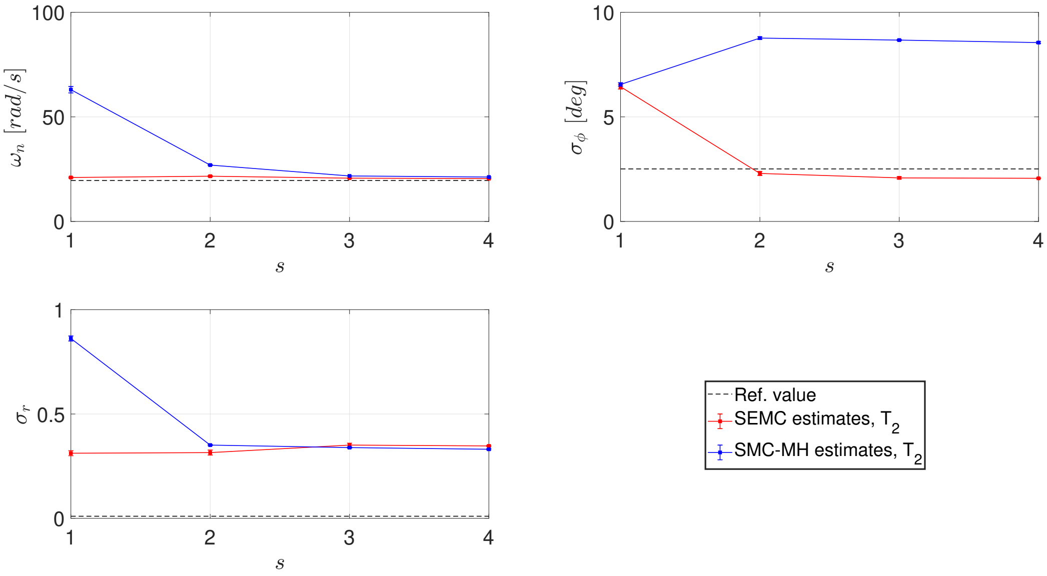


Figure 8.11: Graphical plots summarising the statistics of the parameter estimates of ω_n, σ_ϕ, and σ_r, along with the 3-σ_{SE} bounds, by the SEMC and SMC-MH samplers given the Markov kernel T_2. The reference values for the respective parameters are: $\{\omega_n, \sigma_\phi, \sigma_r\} = \{19.572\,\text{rad/s}, 2.512^o, 0.010\}$.

s	Experimentally identified value	SMC-MH | T_1			SMC-MH | T_2		
		$E[\omega_n]$	σ_{SE}	3-σ_{SE} bounds	$E[\omega_n]$	σ_{SE}	3-σ_{SE} bounds
	$[rad/s]$	$[rad/s]$	$[rad/s]$	$[rad/s]$	$[rad/s]$	$[rad/s]$	$[rad/s]$
1	19.572	46.264	0.440	$[44.944, 47.584]$	63.015	0.513	$[61.476, 64.554]$
2	19.572	21.572	0.192	$[20.996, 22.148]$	26.953	0.089	$[26.686, 27.220]$
3	19.572	21.327	0.126	$[20.949, 21.705]$	21.747	0.027	$[21.666, 21.828]$
4	19.572	20.555	0.080	$[20.315, 20.795]$	21.232	0.018	$[21.178, 21.286]$

Table 8.10: Statistics of the parameter estimates of ω_n by the SMC-MH sampler given each Markov kernel.

8.5 Chapter conclusion

The work presented in the chapter has demonstrated the implementation of the proposed Sequential Ensemble Monte Carlo sampler in performing sequential Bayesian inference on the time-varying and time-invariant parameters simultaneously using measured data under realistic settings. Through the study, the robustness of the proposed sampler is illustrated through its capability in identifying the most probable Markov kernel under model uncertainty as well as being able to control the acceptance rates effectively across the time step

s	Reference value	SEMC | T_1			SEMC | T_2		
		$E[\sigma_\phi]$	σ_{SE}	3-σ_{SE} bounds	$E[\sigma_\phi]$	σ_{SE}	3-σ_{SE} bounds
	$[deg]$	$[deg]$	$[deg]$	$[deg]$	$[deg]$	$[deg]$	$[deg]$
1	2.512	2.841	0.004	$[2.829, 2.853]$	6.442	0.038	$[6.328, 6.556]$
2	2.512	2.786	0.004	$[2.774, 2.798]$	2.295	0.033	$[2.196, 2.394]$
3	2.512	3.133	0.003	$[3.124, 3.142]$	2.082	0.019	$[2.025, 2.139]$
4	2.512	3.282	0.003	$[3.273, 3.291]$	2.062	0.007	$[2.041, 2.083]$

Table 8.11: Statistics of the parameter estimates of σ_ϕ by the SEMC sampler given each Markov kernel.

s	True value	SMC-MH | T_1			SMC-MH | T_2		
		$E[\sigma_\phi]$	σ_{SE}	3-σ_{SE} bounds	$E[\sigma_\phi]$	σ_{SE}	3-σ_{SE} bounds
	$[deg]$	$[deg]$	$[deg]$	$[deg]$	$[deg]$	$[deg]$	$[deg]$
1	2.512	6.400	0.028	$[6.316, 6.484]$	6.550	0.028	$[6.466, 6.634]$
2	2.512	7.519	0.010	$[7.489, 7.549]$	8.767	0.021	$[8.704, 8.830]$
3	2.512	7.692	0.010	$[7.662, 7.722]$	8.671	0.010	$[8.641, 8.701]$
4	2.512	7.715	0.008	$[7.691, 7.739]$	8.552	0.014	$[8.510, 8.594]$

Table 8.12: Statistics of the parameter estimates of σ_ϕ by the SMC-MH sampler given each Markov kernel.

s	Reference value	SEMC | T_1			SEMC | T_2		
		$E[\sigma_r]$	σ_{SE}	3-σ_{SE} bounds	$E[\sigma_r]$	σ_{SE}	3-σ_{SE} bounds
1	0.010	0.313	0.002	$[0.307, 0.319]$	0.312	0.004	$[0.300, 0.324]$
2	0.010	0.329	0.001	$[0.326, 0.332]$	0.315	0.004	$[0.303, 0.3327]$
3	0.010	0.244	0.001	$[0.241, 0.247]$	0.351	0.003	$[0.342, 0.360]$
4	0.010	0.207	0.001	$[0.204, 0.210]$	0.347	0.001	$[0.344, 0.350]$

Table 8.13: Statistics of the parameter estimates of σ_r by the SEMC sampler given each Markov kernel.

sequences whilst inferring both the time-varying and time-invariant parameters.

In addition, the experimental investigation has also shown that the choice of Markov kernel can have an impact on the estimation of the parameters as seen in the case of the time-varying Coulomb friction parameter. This demonstrates the applicability of the proposed sampler in addressing engineering problems.

Furthermore, the effective sample size computed across the time step sequences are generally larger and falls below the threshold value on less occasions for the case of the Se-

s	True value	SMC-MH | T_1			SMC-MH | T_2		
		$E[\sigma_r]$	σ_{SE}	3-σ_{SE} bounds	$E[\sigma_r]$	σ_{SE}	3-σ_{SE} bounds
1	0.010	0.877	0.003	$[0.868, 0.886]$	0.862	0.004	$[0.850, 0.874]$
2	0.010	0.817	0.002	$[0.811, 0.823]$	0.351	0.001	$[0.348, 0.354]$
3	0.010	0.752	0.002	$[0.746, 0.758]$	0.339	0.001	$[0.336, 0.342]$
4	0.010	0.700	0.002	$[0.694, 0.706]$	0.331	0.001	$[0.328, 0.334]$

Table 8.14: Statistics of the parameter estimates of σ_r by the SMC-MH sampler given each Markov kernel.

quential Ensemble Monte Carlo sampler compared to the Sequential Monte Carlo sampler with Metropolis-Hastings kernel. This indicates a better mixing performance by the Affine-invariant Ensemble sampler which justifies its use as a relatively efficient MCMC move kernel than the MH sampler.

Finally, the MATLAB codes for the example presented in the chapter is made accessible on GitHub: `https://github.com/Adolphus8/Sequential_Ensemble_Monte_Carlo.git`

Chapter 9

Probabilistic Prediction of Nuclear Material Properties with Artificial Intelligence under Uncertainty

Abstract

The chapter presents a feasibility study aimed at developing a framework for the Probabilistic Artificial Intelligence Prediction of Material Properties for Nuclear reactor designs. Currently, Artificial Intelligence approaches are not largely implemented in the nuclear sector compared to other sectors such as aerospace and manufacturing. The key challenge is the availability of sparse data set to train AI models and the poor consideration of the uncertainty in the data and the model. As such, the study seeks to propose a stochastic approach towards enhancing the sparse data set from which the Bayesian inference framework is implemented to account for the model uncertainty in the prediction of the material properties. Such proposal involves increasing the training data set from the initial sparse data set while retaining the physical dependencies among variables. This allows for a larger data set to train a set of Artificial Neural Networks used to predict selected nuclear material properties. From there, the Adaptive Bayesian Model Selection method is implemented to account for the variability in the prediction results due to the model uncertainty and quantifies such uncertainty using the Bayesian inference framework. This allows for robust predictions along with the associated confidence bounds. The resulting estimates obtained using the proposed approach are well-validated against the experimental data with improved accuracy.

9.1 Background

This work presents a feasibility study aimed at devising a framework towards providing a probabilistic Artificial Intelligence (AI) prediction of material properties relevant for the design of Nuclear reactors.

At present, the existing AI models are trained using deterministic data set without accounting for the uncertainties associated with the measured quantity. As such, a large number of physical tests and experimental campaigns are required to increase the training data size and account for the inherent variability of the material properties. However, doing so would also increase the research costs. In a study by [320], each fracture test on a material can amount to £15000 per test. Furthermore, an AI model is assumed to characterise relationship between the key input and output features of interest. In general, such assumption may not hold true given the lack of certainty over the "true" relationship between the input and output features, especially so in the absence of a physics-based model.

Hence, this work seeks to realize 3 main research objectives: 1) to improve the robustness of the estimates by considering the inherent uncertainty on the training (i.e. experimental) data set; 2) to enable the Artificial Neural Networks (ANNs) to deal with sparse and "noisy" data in providing estimates on the quantity of interest along with its associated confidence bounds; and 3) to loosen the assumption on the choice of model by considering a set of ANNs, with different architectures, constructed to predict the same quantity of interest (i.e. model uncertainty).

9.2 State of Artificial Intelligence in Nuclear

In a recent study, it was found that the AI technology is not widely implemented within the Nuclear sector and that the sector is currently lagging behind in the Industry 4.0 revolution compared to other industries such as the healthcare, automotive and manufacturing [320]. In addition, the industry seeks to develop new reactor designs for both fusion and fission. These would come in the form of the newer generation Light Water Reactors, Liquid Metal Cooled Reactors, and High-temperature Gas-cooled Reactors which are expected to be more modular and compact in their designs. Furthermore, the nuclear industry has to deal with 2 key challenges [320]: 1) the need to decommission ageing reactors; and 2) the high costs of building new reactors. With the latter point, it makes nuclear energy a much more expensive option compared to other energy sources such as wind and solar. In this regard, this brings opportunities for the application of AI which can be expected to play an important role in devising new ways to design, construct, operate, and decommission such reactors across the entire project operation duration.

The role of AI within the Nuclear sector can be categorised into 2 types [320]: 1) Decentralised decision-making; and 2) Technical assistance. The first category refers to the capability of the AI systems to become autonomous, thereby being able to make simple decisions itself without human intervention. The second category refers to the capability of the AI systems in supporting humans in 2 aspects: 1) the decision-making process towards solving a problem; and 2) providing assistance in completing tasks which are too complex and risky for humans. These include accident identification, system performance, structural integrity, predictive maintenance, and predicting material properties and behaviours. The focus of the chapter is on the technical assistance aspect of AI in the context of material properties prediction.

9.3 Methodology

For this feasibility study, a database of the material properties for 58 different steel types is obtained from the National Institute for Materials Science (NIMS) based on a previous experimental campaign under the Material Properties Predictor for Power Plant Steels (M4PS) project ([321]). The material properties covered in this database includes:

- Creep rupture properties - with 8005 observations;
- Tensile properties - with 2878 observations;
- Hardness properties - with 234 observations.

For the work presented here, the main focus would be on the prediction of Creep rupture and Tensile properties of the steel materials whose key input and target features are summarised in Tables 9.1 and 9.2 respectively. As an illustration, the scatterplot diagrams of the corresponding raw data for the selected input and target features of the Creep rupture and Tensile properties are presented in Figures 9.1 and 9.2.

From Figures 9.1 and 9.2, it can be observed from the scatterplot profiles of the data that there are significant gaps between the data points. In addition, it can also be seen that some data points are grouped about discrete values as seen in the plot for Elongation vs Temperature in Figure 9.1 and that for RA vs Temperature in Figure 9.2. This leads

Input features	Target features
Material code	Fracture Time (FT)
Cast code	Elongation
Stress	Reduction of Area (RA)
Temperature	
Composition (19 elements)	

Table 9.1: List of input and target features of interest for Creep rupture properties prediction.

Input features	Target features
Material code	Ultimate Tensile Strength (UTS)
Cast code	Elongation
Temperature	0.2 % Proof Stress (PS02)
Composition (19 elements)	RA

Table 9.2: List of input and target features of interest for Tensile properties prediction.

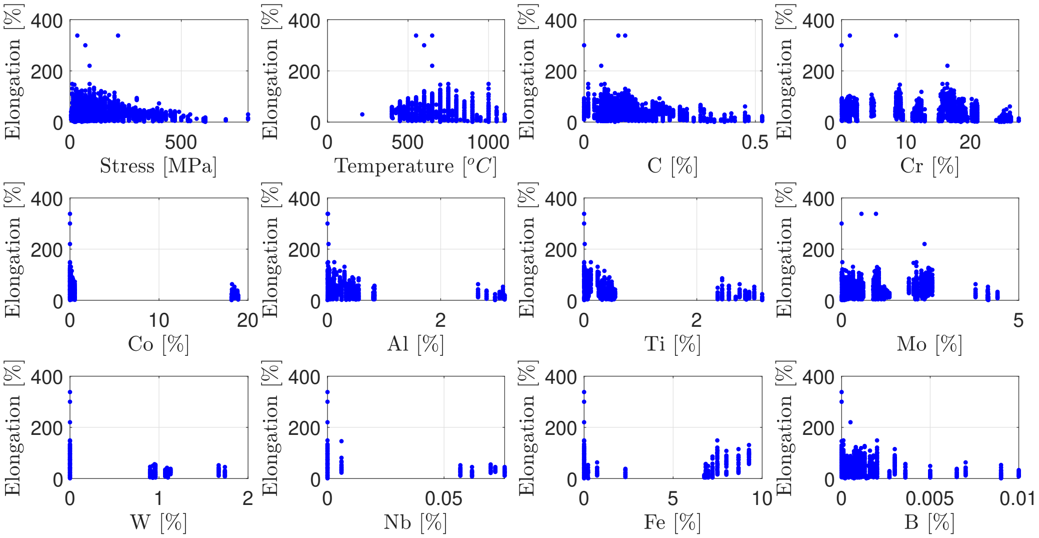


Figure 9.1: Scatterplot representation of the raw data of the selected input features (i.e. Stress, Temperature, and some of the comprising elements) for Creep rupture properties against the Elongation target feature.

to significant loss of information as the data points do not explore the entire domain of the experimental input values. As such, the need to enhance the data set arises to fill such gaps so as to be able to effectively train the ANNs subsequently. A way to perform such enhancement without he need to perform additional experimental campaigns would be

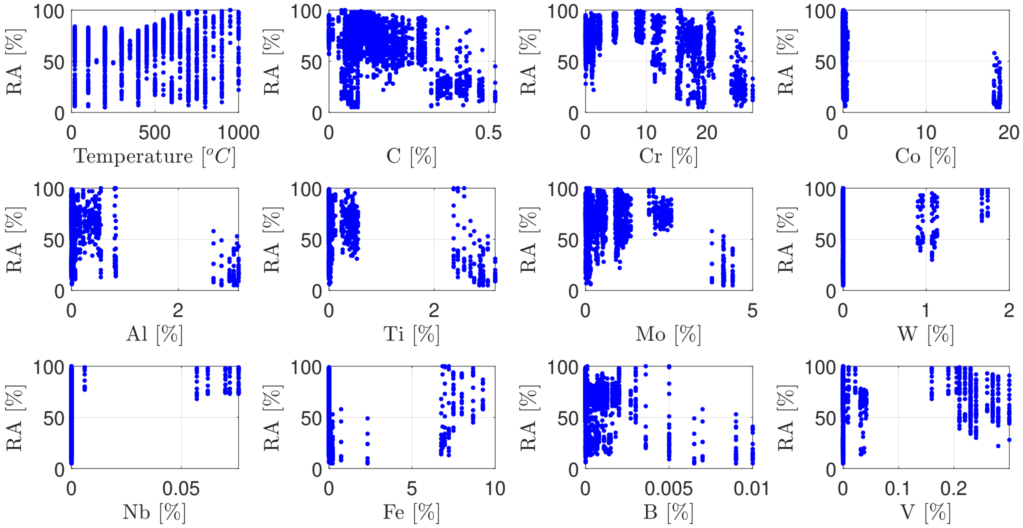


Figure 9.2: Scatterplot representation of the raw data of the selected input features (i.e. Stress, and some of the comprising elements) for Tensile properties against the RA target feature.

through a stochastic approach to which details are presented in Section 9.3.1.

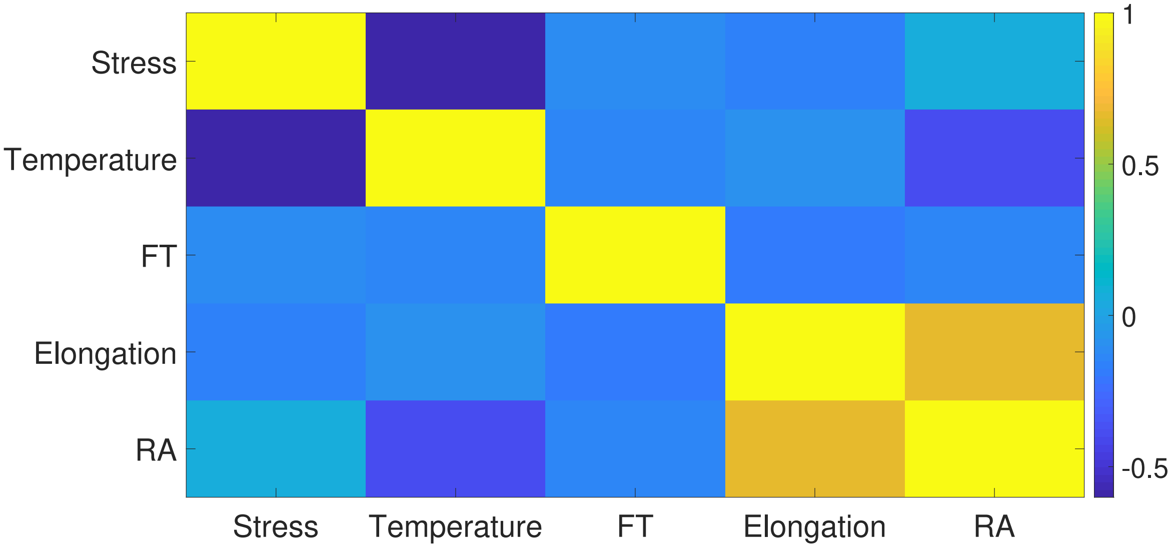
9.3.1 Data Enhancement

In this study, a particular category of steel type is chosen for Creep rupture and Tensile properties prediction. This leaves 13 observations for Creep rupture properties and 8 observations for Tensile properties from the initial experimental data set.

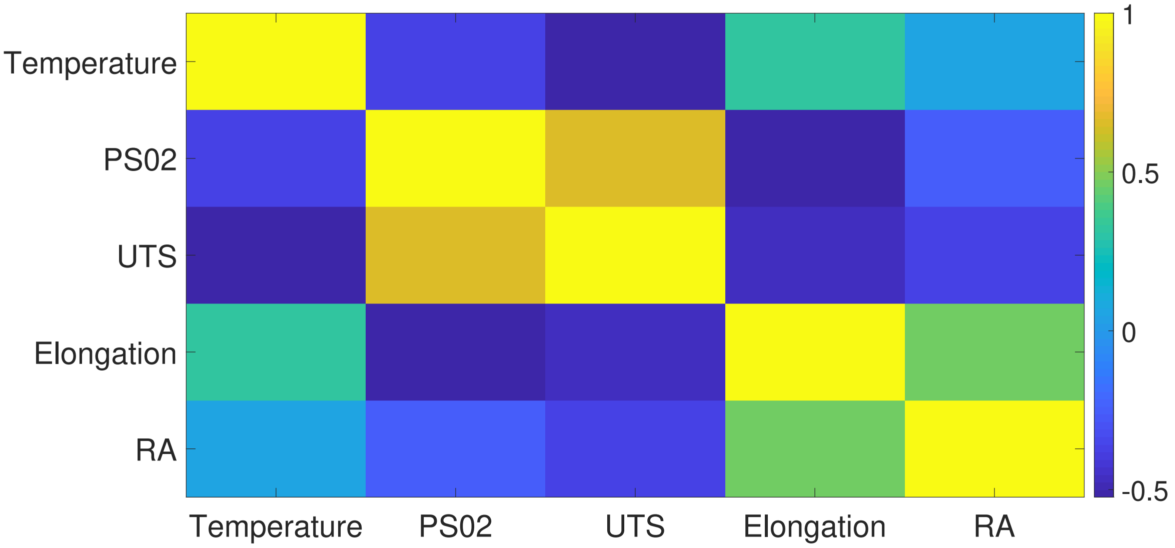
To capture the physical relations between the key input and target features in the absence of any physics-based model, the Pearson correlation coefficient ρ between the m^{th} and n^{th} features is computed [322]:

$$\rho = \frac{\sum_{i=1}^{N_{data}} (y_i^m - \bar{y^m}) \cdot (y_i^n - \bar{y^n})}{\sqrt{\sum_{i=1}^{N_{data}} (y_i^m - \bar{y^m})^2 \cdot \sum_{i=1}^{N_{data}} (y_i^n - \bar{y^n})^2}} \tag{9.1}$$

where N_{data} is the data size, y_i^m and y_i^n are the i^{th} experimental data-point of the m^{th} and n^{th} feature respectively, while $\bar{y^m}$ and $\bar{y^n}$ are the mean experimental values of the m^{th} and n^{th} feature respectively. Hence, $\rho = 1$ for the case when $m = n$. This procedure is done for the Creep rupture and Tensile properties data to which the resulting correlation matrix for the selected features of the Creep rupture and Tensile data are illustrated in Figure 9.3 while the numerical values to the Pearson correlation coefficient between the selected input and target features of the Creep rupture and Tensile properties are presented in Tables 9.3 and 9.4 respectively.



(a)



(b)

Figure 9.3: Colourplot matrix illustrating the correlation between the selected input and target features of the: (a) Creep rupture properties; (b) Tensile properties.

	Stress	Temperature	FT	Elongation	RA
Stress	1	-0.601	-0.117	-0.154	0.053
Temperature	-0.601	1	-0.149	-0.095	-0.376
FT	-0.117	-0.149	1	-0.180	-0.141
Elongation	-0.154	-0.095	-0.180	1	0.659
RA	0.053	-0.376	-0.141	0.659	1

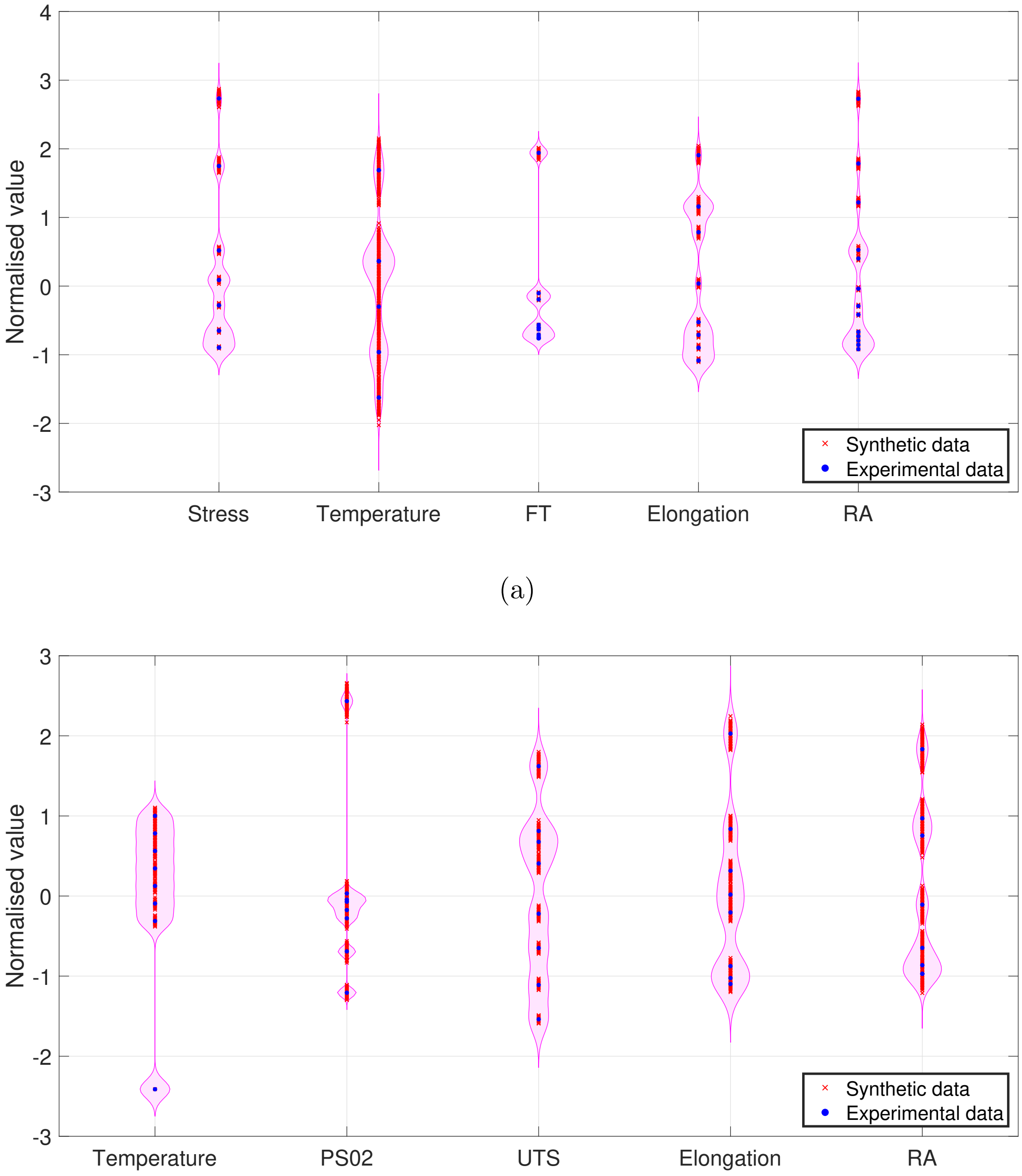
Table 9.3: Numerical values to the Pearson correlation coefficient between the selected input and target features of the Creep rupture properties.

	Temperature	PS02	UTS	Elongation	RA
Temperature	1	-0.371	-0.508	0.322	0.052
PS02	-0.371	1	0.661	-0.524	-0.240
UTS	-0.508	0.661	1	-0.467	-0.360
Elongation	0.322	-0.524	-0.467	1	0.470
RA	0.052	-0.240	-0.360	0.470	1

Table 9.4: Numerical values to the Pearson correlation coefficient between the selected input and target features of the Tensile properties.

Following this, a correlated multivariate Gaussian distribution is generated whose covariance matrix is constructed such that the standard deviation of each feature is set at 1 % of its data value while the correlation between the features is based on the computed ρ. When this is done, the correlated multivariate Gaussian distribution is then centered about a chosen experimental data from which a set of synthetic data, defined by the analyst, is generated. Such procedure is performed across all the experimental data such that a total of $N_{data} = 10000$ synthetic data is obtained. This is done for both the Creep rupture and Tensile properties data to which the resulting violin plots are illustrated in Figure 9.4. For the illustrative purpose of presenting the data-points in the same scale, the data values of the corresponding features have been standardised according to the standard normal in the aforementioned figures.

It needs to be highlighted that the use of a multivariate Gaussian distribution model to generate the synthetic data set is done under the assumption that the measurement noise follows a zero-mean Normal distribution. In general, such assumption may not be true and and the selection of an appropriate stochastic model to generate the synthetic data set introduces a significant degree of model uncertainty.



(a)

(b)

Figure 9.4: Violin plot of the synthetic data along with the experimental data for the following material property prediction: (a) Creep rupture properties; (b) Tensile properties.

9.3.2 Artificial Neural Network training

Due to the absence of a physics-based mathematical model describing the relationship between the input and target features, ANNs are used as the surrogate models due to its effectiveness and versatility in modelling such relationships. Descriptions on the concept of ANNs can be found in Section 5.2.

In this study, the specific type of ANN used is the Feed-forward Neural Network [259] with the rectified linear unit activation function [323] being the activation function. For both Creep rupture and Tensile properties, 5 of such ANNs are constructed to predict the corresponding target features. This is to loosen the assumption on the choice of the model and introduce elements of model uncertainty. Each of these ANNs have one output node but differ in the number of hidden-layers and/or hidden-nodes and are trained using the synthetic data presented in Section 9.3.1. In training the ANNs, the 10000 sets of synthetic data are split in the following manner: 70 % was used to train the ANN, 15 % for validation, and 15 % for testing. The resulting training times for the corresponding ANN model to predict the target features for Creep rupture and Tensile properties are presented in Tables 9.5 and 9.6 respectively.

From there, the trained ANNs are validated with the experimental data used to generate the synthetic data. The performance index used to quantify the robustness of the ANN model validation against the experimental data, for a given m^{th} target feature, is the R^2-score defined as [324]:

$$R^2 = \left(1 - \frac{\sum_{i=1}^{N_{data}}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N_{data}}(y_i - \bar{y}_i)^2}\right) \times 100\% \tag{9.2}$$

where $\hat{y}_i$ is the ANN model prediction of the i^{th} data-point of the target feature of interest. The resulting R^2-score computed for the corresponding ANN model in its prediction of the target features for Creep rupture and Tensile properties, relative to the experimental data, are presented in Tables 9.5 and 9.6 respectively.

Model no.	Configuration	Time [s]			R^2-score [%]		
		FT	Elongation	RA	FT	Elongation	RA
1	23 – 18 – 1	1.63	3.58	3.48	99.85	99.98	99.98
2	23 – 32 – 1	–	3.27	1.65	–	99.94	99.95
3	23 – 18 – 9 – 1	5.29	10.42	7.60	99.99	99.99	99.99
4	23 – 27 – 18 – 9 – 1	17.79	18.04	119.68	99.99	99.99	99.99
5	23 – 64 – 32 – 8 – 1	98.14	–	–	99.99	–	–
6	23 – 64 – 32 – 16 – 1	540.94	314.35	298.52	99.99	99.99	99.99

Table 9.5: Training times and R^2-score of the respective ANN model for the target features for Creep rupture property prediction.

From Tables 9.5 and 9.6, it can be seen that the R^2-scores are within the interval of

Model no.	Configuration	Time [s]				R^2-score [%]			
		UTS	Elongation	PS02	RA	UTS	Elongation	PS02	RA
1	$22-18-1$	28.97	60.70	20.42	4.11	94.68	99.88	96.60	94.58
2	$22-32-1$	$-$	$-$	7.09	$-$	$-$	$-$	96.59	$-$
3	$22-64-1$	45.14	5.14	$-$	6.56	99.99	99.99	$-$	94.58
4	$22-18-9-1$	28.45	11.54	17.04	21.67	99.84	99.88	92.91	94.55
5	$22-27-18-9-1$	92.49	356.39	34.85	127.47	99.97	99.99	92.94	94.82
6	$22-64-32-16-1$	4180.60	1286.30	649.41	1534.80	99.99	100.00	99.98	99.95

Table 9.6: Training times and R^2-score of the respective ANN model for the target features for Tensile property prediction.

$[92.91, 100.0]$. This indicates that the trained ANN models are well-validated against the experimental data and highlights the effectiveness of the use of the synthetic data as training data. The reason for such result is due to the fact that the synthetic data includes the values of the experimental data, ensuring that the data region where the experimental data exists is also included in the training of the ANN.

Further investigations are also being done to study the robustness of the ANNs constructed for this work. In this study, the ANNs for the prediction of the UTS property are trained with the entire experimental data set from NIMS. From there, the ANNs are verified against the experimental data where it was found that despite a potential error in the recorded data, the ANNs do not take into account such error. As an illustrative example, a graphical plot of the result is provided in Figure 9.5 for the ANN with configuration: $22 - 64 - 32 - 16 - 1$. This further justifies the use of the Feed-forward Neural Network as the choice of ANN for this study. It needs to be noted, however, that such robustness is observed due to there being only one outlier in the data. Should there be multiple outliers within the trained data set, such results may not be observed and the ANN might eventually predict such outliers.

9.3.3 Adaptive Bayesian Model Selection

To provide a robust probabilistic prediction of a given target feature by a set of ANNs with the associated confidence bounds, the Adaptive Bayesian Model Selection (ABMS) method, proposed by [325], is implemented. It uses an adaptive Bayesian model selection approach to compute the continuous posterior distribution over a set of N_m distinct ANN models of different configurations.

For a given v^{th} ANN model M_v (for $v = 1, \dots, N_m$) and training data set: $\boldsymbol{D} =$

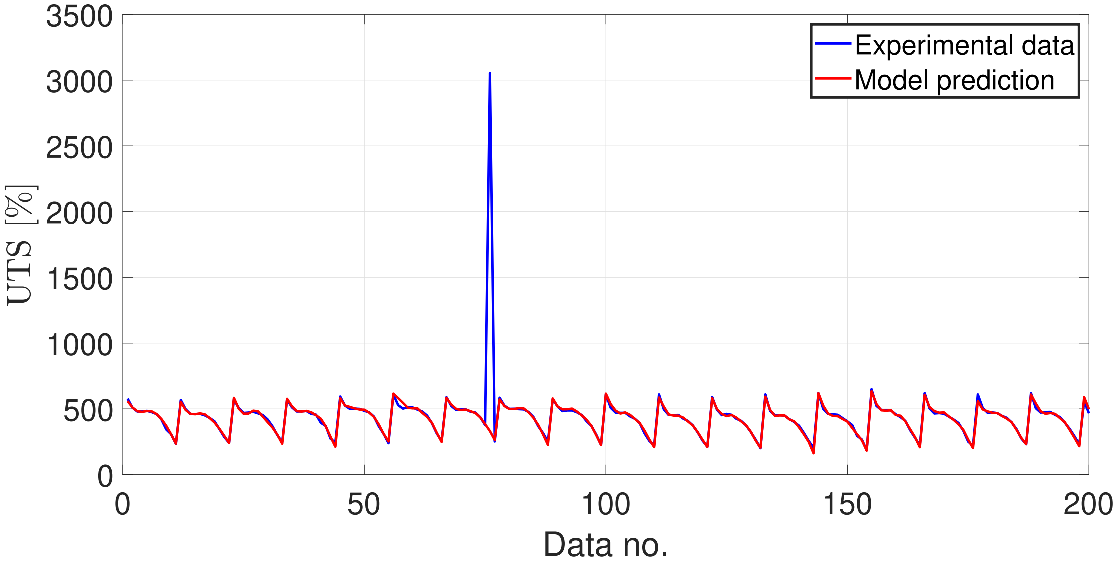


Figure 9.5: ANN prediction of UTS against the experimental data with configuration: $22 - 64 - 32 - 16 - 1$.

$\{(x_1, y_1), \ldots, (x_{N_{data}}, y_{N_{data}})\}$, its continuous posterior probability $P(M_v, y_v|\boldsymbol{D})$ is defined according to Bayes' theorem [46, 236]:

$$P(M_v, y_v|\boldsymbol{D}) = \frac{P(\boldsymbol{D}|M_v, y_v) \cdot P(M_v, y_v)}{\sum_{u=1}^{N_m} P(\boldsymbol{D}|M_u, y_u) \cdot P(M_u, y_u)} \tag{9.3}$$

where x_i is the i^{th} input feature data, for $i = 1, \ldots, N_{data}$, $P(M_v, y_v)$ is the prior, $P(\boldsymbol{D}|M_v, y_v)$ is the likelihood function, the denominator term is the normalisation constant, and $y_v = \hat{y}_v + \epsilon_v$ is the true prediction value for which $\epsilon_v \sim N(0, {\sigma_v}^2)$ is the bias associated with the measurement "noise" and model error. The higher the value of $P(M_v, y_v|\boldsymbol{D})$, the more reliable the value of y_v for the given ANN model M_v.

The computation of the ABMS robust prediction y_{rb} follows: Considering the ANN prediction $\hat{y}_v$ from M_v, it has an associated posterior probability $P(M_v, \hat{y}_v|\boldsymbol{D})$. As such, the best model prediction $\hat{y}^*$, defined as the prediction with the highest degree of accuracy, has an associated posterior probability $\max\left[P(M_v, \hat{y}^*|\boldsymbol{D})\right]$, for $v = 1, \ldots, N_m$. From there, y_{rb} is computed [325]:

$$y_{rb} = \hat{y}^* + \sum_{v=1}^{N_m} P(M_v, \hat{y}_v|\boldsymbol{D}) \cdot (\hat{y}_v - \hat{y}^*) \tag{9.4}$$

To derive the 95 % confidence bounds on y_{rb}, denoted as $[\underline{y_{rb}}, \overline{y_{rb}}]$, the standard deviation term Φ is first computed:

$$\Phi = \sqrt{\sum_{v=1}^{N_m} P(M_v, \hat{y}_v | \boldsymbol{D}) \cdot (\hat{y}_v - y_{rb})^2} \tag{9.5}$$

Upon which, the corresponding values of $\underline{y_{rb}}$ and $\overline{y_{rb}}$ are calculated as follows:

$$\left[\underline{y_{rb}}, \overline{y_{rb}}\right] = y_{rb} \mp 1.95 \cdot \Phi \tag{9.6}$$

Further details on the mathematical derivations and the ABMS algorithm are found in [325].

For the analysis, the prior $P(M_v, y_v)$ is set as a Uniform distribution given the apriori assumption that all N_m ANN models are equally likely. The likelihood function for a given i^{th} data sample (x_i, y_i) is defined as ([325]):

$$P((x_i, y_i)|M_v, \hat{y}_{v,i}) = \frac{1}{\sigma_v \cdot \sqrt{2\pi}} \cdot \exp\left[-\frac{(y_i - \hat{y}_{v,i})}{2 \cdot {\sigma_v}^2}\right] \tag{9.7}$$

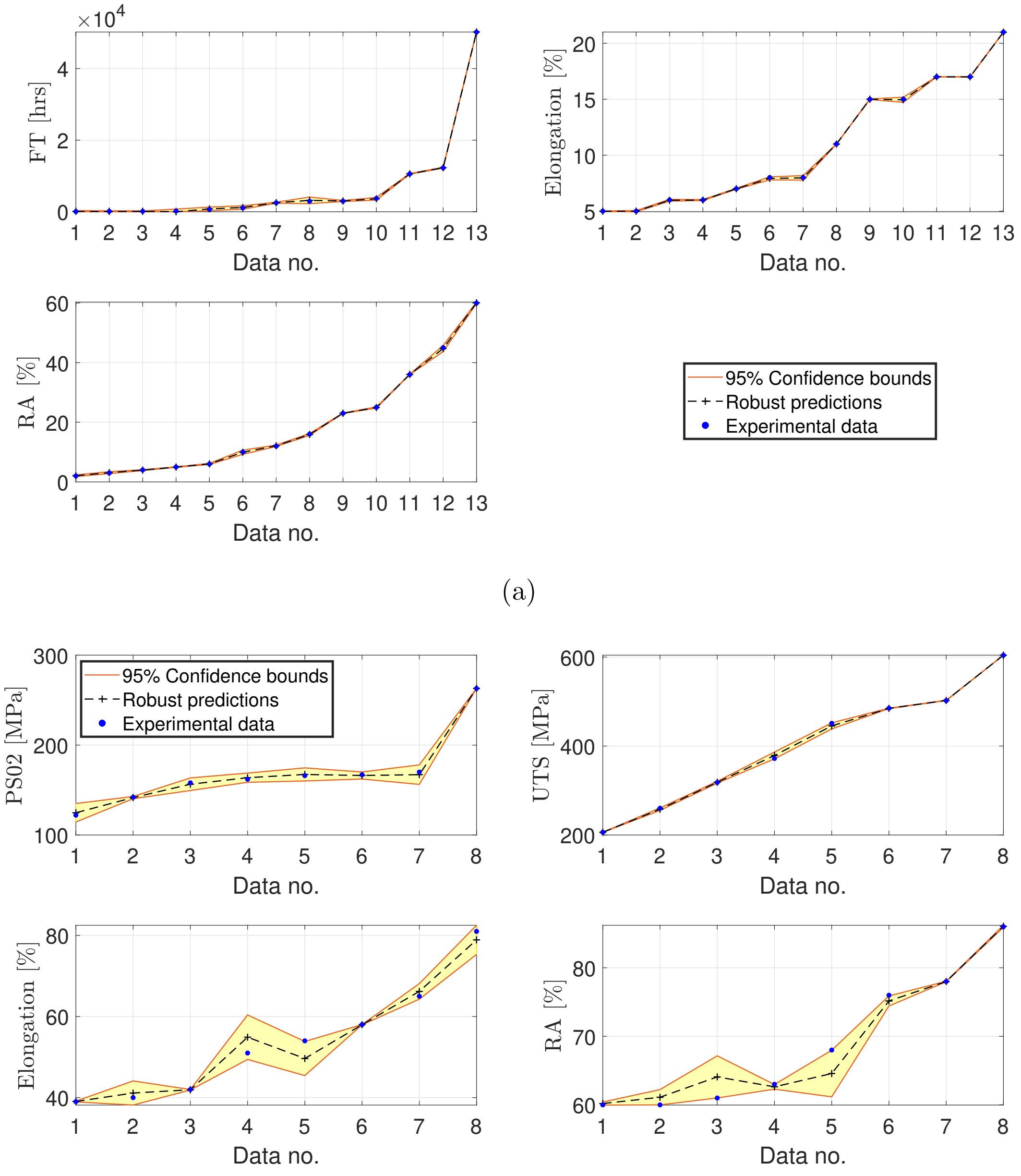
where σ_k can be estimated from $\boldsymbol{D}$ using maximum likelihood estimation. The posterior $P(M_v, y_v|\boldsymbol{D})$ is approximated using a Gaussian Mixture Model. For each target feature to be predicted, $N_m = 5$ ANNs are used as listed in Tables 9.5 and 9.6. The synthetic data presented in Section 9.3.1 would serve as the training data $\boldsymbol{D}$ while the experimental data is used to validate the prediction results from ABMS.

9.4 Results and discussions

The resulting probabilistic prediction of the steel's Creep rupture and Tensile properties are presented as graphical plots in Figure 9.6. From the figure, it can be seen that the 95 % confidence bounds generally encloses all the experimental data-points with all of such points falling within or on either bounds. This indicates that the probabilistic prediction by the ABMS method is well-validated against the experimental data which can be justified from the high R^2-scores (i.e. above 90 %) of the ANNs used to predict the respective target features as seen in Tables 9.5 and 9.6.

In addition, it can be observed from Figure 9.6(b) that the 95 % confidence bounds are significantly wider for the Elongation and RA prediction compared to those for the same target features in Figure 9.6(a). A reason for this is due to the worst R^2-scores for the ANN Tensile property predictions of Elongation and RA is at 99.88 % and 94.55 % respectively.

Such scores are significantly lower in comparison to that for the Creep rupture property predictions of Elongation and RA which are at 99.94 % and 99.95 % respectively. This gives rise to a lower-degree of precision, leading to wider bounds seen in Figure 9.6(b) for the aforementioned target feature predictions.



(a)

(b)

Figure 9.6: Robust predictions, along with its 95 % confidence bounds, of the target features of the chosen steel material's: (a) Creep rupture properties; (b) Tensile properties.

The computation times by the ABMS algorithm for the prediction of the target features of the corresponding steel material properties are presented in Table 9.7. From the table, it can be observed that the computation time of the ABMS is generally longer for the case of predicting the Creep rupture than Tensile properties of the steel of interest. This is because, there are more experimental data points of the input feature in the case of predicting the Creep rupture properties (i.e. 13 experimental data points) compared to the prediction of the Tensile properties (i.e. 8 data points).

Material property	Target feature	Time $[s]$
Creep rupture	FT	6.33
	Elongation	4.51
	RA	5.38
Tensile	PS02	2.69
	UTS	2.36
	Elongation	2.01
	RA	2.71

Table 9.7: Computation times for the respective target feature of a given material property of the steel studied.

Finally, the resulting posterior probability $P(M_v, y_v|\boldsymbol{D})$ (i.e. see Eq. (9.3)) associated with each of the ANN models in predicting the Creep rupture and Tensile properties of the material are presented graphically in Figures 9.7 and 9.8 respectively. For both figures, the colours correspond to the following ANN models: Red for ANN model 1; Blue for ANN model 2; Green for ANN model 3; Magenta for ANN model 4; Yellow for ANN model 5; and Cyan for ANN model 6. Details on the ANN configuration for the corresponding model number are found in Tables 9.5 and 9.6 for Creep rupture and Tensile properties prediction respectively. It needs to be highlighted that in the prediction of the Elongation and Reduction of Area (RA) of the material in Figure 9.8, the ANN models have almost equal posterior probability $P(M_v, y_v|\boldsymbol{D})$.

9.5 Chapter conclusion

The work presents a general framework towards a robust probabilistic prediction on the Nuclear material properties under sparse data. It involves the stochastic enhancement of the experimental data set explained in Section 9.3.1, followed by the construction and training of a set of ANNs with the synthetic data seen in Section 9.3.2, before finally incorporating

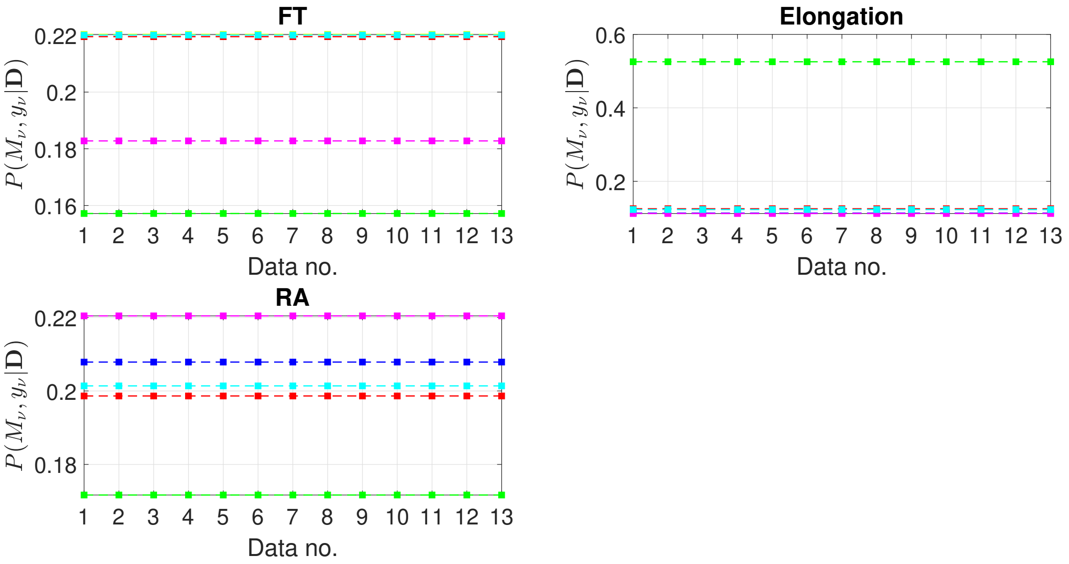


Figure 9.7: Graphical plot of the resulting posterior probability of each of the given ANN model in predicting the corresponding data points associated with the Creep rupture properties of the material.

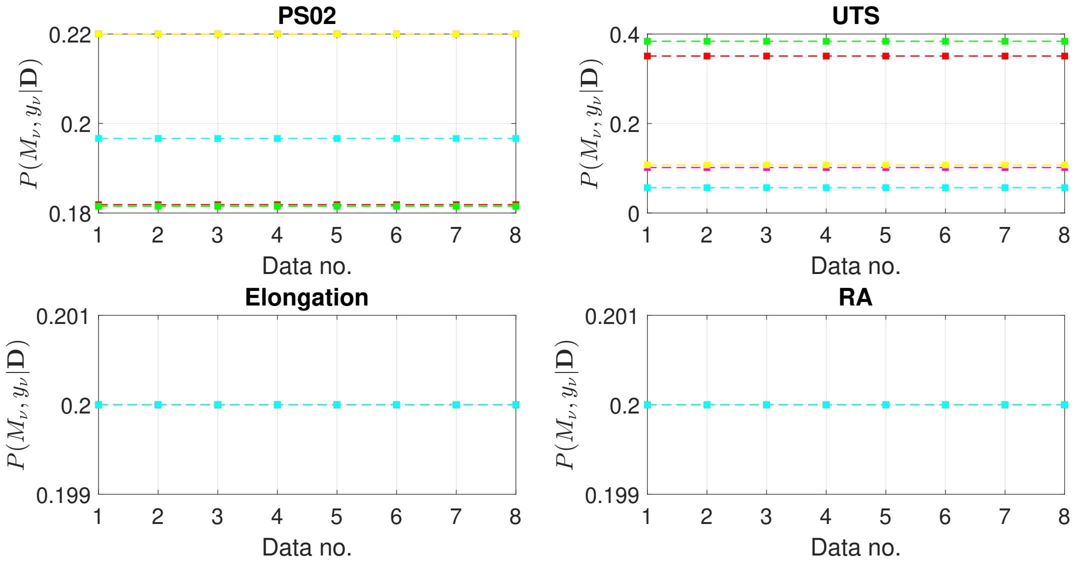


Figure 9.8: Graphical plot of the resulting posterior probability of each of the given ANN model in predicting the corresponding data points associated with the Tensile properties of the material.

Bayesian statistics in the ANN predictions to yield probabilistic estimates via ABMS which are well-validated against the experimental data as shown in Section 9.4.

There are 4 key benefits which this study seeks to provide: 1) by providing a probabilistic prediction instead of a deterministic one, the uncertainty of the estimates is accounted for. This allows for the users to determine the level of confidence on the predictions as well as make an informative risk-based decision on the choice of materials to use in the design of new Nuclear reactors; 2) the proposed framework accounts for the uncertainty associated with the choice of the ANN models used for the prediction of the material properties. This was done by considering 5 different ANN models to predict each target feature, thereby loosening the assumption on the choice of models used. From which, a robust prediction is provided based on Adaptive Bayesian Model Selection and Model Averaging along with the 95 % confidence bounds to account for the model error; 3) the proposed framework addresses the issue of sparse data via the stochastic data-enhancement method where the use of a Gaussian Mixture Model, along with the information on the correlation between the features of interest, is used to generate synthetic data from the experimental data. It needs to be re-iterated, however, that the approach is implemented under the assumption that the measurement noise follows a zero-mean Normal distribution and that such assumption may not necessarily be true at all times. As such, broadly speaking, the selection of an appropriate stochastic model to generate the synthetic data set introduces a significant degree of model uncertainty; and 4) this framework can help reduce the need to run multiple experimental campaigns, thereby saving costs.

For the benefit of the readers in understanding the proposed framework as well as to reproduce the results presented in the chapter, codes and algorithms used for the study are made available on GitHub via: `https://github.com/Adolphus8/Project_PROMAP.git`

Finally, as a supplement to the work presented in the chapter, a numerical study is presented in the chapter Appendix (i.e. see Section 9.6) which seeks to further illustrate and demonstrate the feasibility of the proposed framework.

9.6 Chapter Appendix

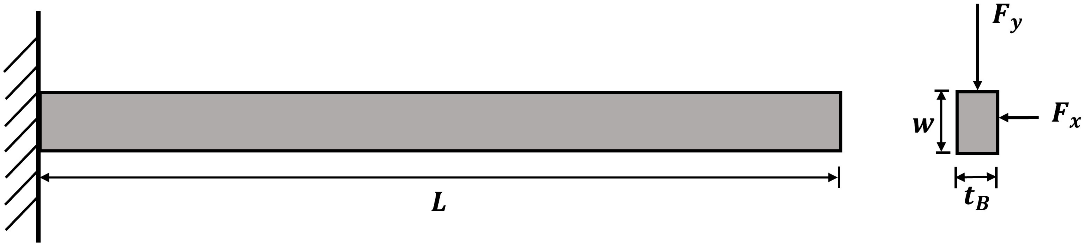


Figure 9.9: Schematic diagram of the Cantilever beam set-up used in the study.

The presented study is a simple regression problem involving a Cantilever beam set-up illustrated in Figure 9.9 which can be used to model the vibration of the secondary core support pillar within a Nuclear Power Plant [326]. The resulting free-end displacement of the beam D can be modelled as follows [327]:

$$D = \frac{4 \cdot L^3}{E \cdot w \cdot t_B} \cdot \sqrt{\left(\frac{F_y}{{t_B}^2}\right)^2 + \left(\frac{F_x}{w^2}\right)^2} \tag{9.8}$$

The descriptions to the respective parameters of Eq. (9.8) are presented in Table 9.8. It needs to be noted that the parameters L, w, t_B, and E are fixed while F_x and F_y are variables.

Parameter	Description	Value(s)	Units
L	Length of the Cantilever beam	2.50	$[m]$
w	Width of the Cantilever beam	0.15	$[m]$
t_B	Thickness of the Cantilever beam	0.20	$[m]$
E	Young's Modulus of the Cantilever beam material	2.00×10^9	$[N/m^2]$
F_x	Applied horizontal load	$[800, 900] \cup [1000, 1200]$	$[N]$
F_y	Applied vertical load	$[300, 500] \cup [600, 700]$	$[N]$

Table 9.8: Descriptions to the key parameters of the Cantilever beam set-up.

9.6.1 Generating the Numerical data

The variables F_x and F_y are sampled independently from a $2D$ joint Uniform distribution with the respective bounds defined in Table 9.8. For each realization of F_x and F_y, the nominal free-end displacement value D_{nom} is obtained via Eq. (9.8). From the value of D_{nom}, a measurement "noise" term ϵ_D is added following:

$$D = D_{nom} + \epsilon_D \tag{9.9}$$

where ϵ_D follows a zero-mean Normal distribution with standard deviation $\sigma_D = 0.0003\ m$. In total, 30 sets of data for F_x, F_y, and D are generated whereby 10 of which will serve as the observed data set while the remaining 20 will serve as the validation data set. The numerical values of the respective data set are presented in Table 9.9.

S/N	Observed data set			Validation data set		
	F_x $[N]$	F_y $[N]$	D $[m]$	F_x $[N]$	F_y $[N]$	D $[m]$
1	834.84	386.29	0.0283	825.55	323.76	0.0263
2	819.91	396.65	0.0283	848.57	327.51	0.0263
3	849.90	422.88	0.0302	871.76	321.57	0.0273
4	872.06	454.28	0.0312	879.54	336.68	0.0276
5	1091.65	314.47	0.0317	829.98	387.39	0.0277
6	1102.16	358.73	0.0327	1021.41	317.62	0.0304
7	1123.39	431.10	0.0364	882.17	436.00	0.0308
8	820.96	643.49	0.0366	1027.93	337.93	0.0312
9	837.70	644.75	0.0371	1088.10	327.91	0.0318
10	1170.27	604.45	0.0413	1070.44	363.23	0.0323
11				803.96	600.89	0.0348
12				1174.54	408.72	0.0354
13				892.75	612.67	0.0365
14				880.61	614.15	0.0365
15				825.10	662.86	0.0377
16				860.16	669.23	0.0385
17				1148.12	613.26	0.0412
18				1095.51	663.07	0.0414
19				1132.75	630.14	0.0416
20				1105.89	677.04	0.0425

Table 9.9: Numerical values to the corresponding data set for F_x, F_y, and D.

As outlined in Section 9.3.1, to ensure that the synthetic data set generated from the observed data set retains the information pertaining to the physical relationship between the input and the output variables, the Pearson correlation coefficient ρ is computed based on the observed data set following Eq. (9.1) The resulting correlation matrix between the variables F_x, F_y, and D are illustrated in Figure 9.10 while the numerical values of the Pearson correlation coefficient ρ between each pair of variables are presented in Table 9.10.

Following which, 3000 sets of synthetic data are generated stochastically from the given observed data set following the methodology outlined in Section 9.3.1 and the resulting violin plot is illustrated in Figure 9.11. For the illustrative purpose of presenting the data-points

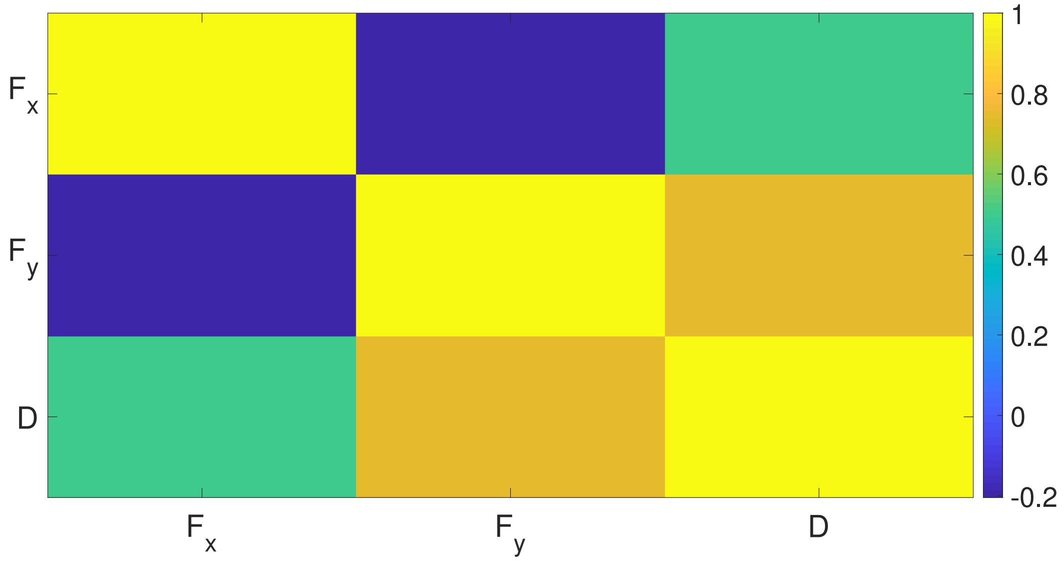


Figure 9.10: Colourplot matrix illustrating the correlation between the variables F_x, F_y, and D.

	F_x	F_y	D
F_x	1	-0.204	0.496
F_y	-0.204	1	0.743
D	0.496	0.743	1

Table 9.10: Numerical values to the Pearson correlation coefficient between F_x, F_y, and D.

in the same scale, the data values of the corresponding variables have been standardised according to the standard normal in the aforementioned figures.

9.6.2 Artificial Neural Network training

For the numerical study, the type of ANN used is the Feed-forward Neural Network with the sigmoid function [258] being the activation function. To loosen the assumption on the choice of the model and introduce model uncertainty, 5 different configurations of the ANN architecture are constructed as presented in Table 9.11 to predict the Cantilever beam's free-end displacement D. It needs to be noted that for this study, small ANN architectures are used given that there are only 2 input variables and one output variable. As outlined in Section 9.3.2, the synthetic data set is used to train the ANNs. In training the ANNs, the synthetic data set is split in the following manner: 60 % was used to train the ANN, 20 % for testing, and 20 % for validation. Once trained, each of the ANNs are validated against

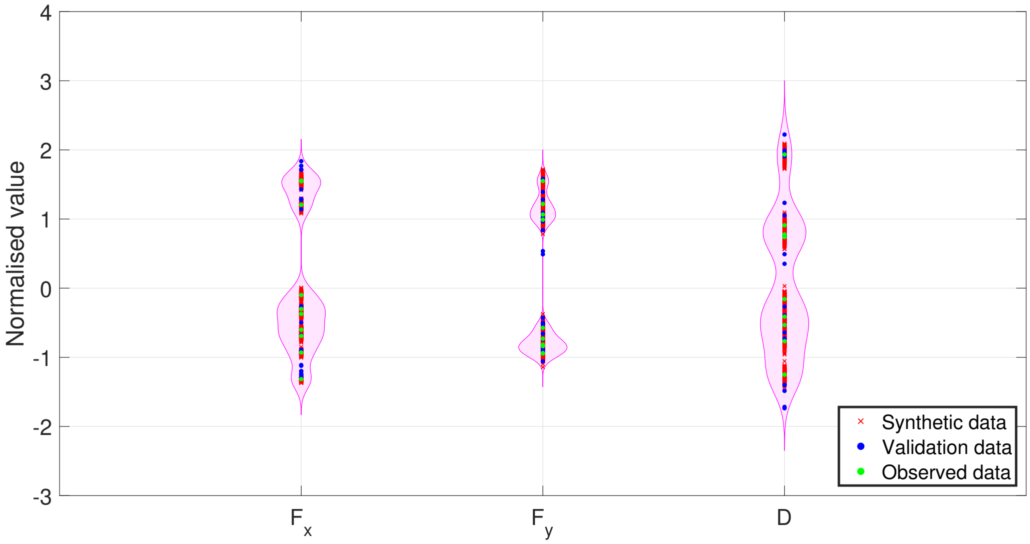


Figure 9.11: Violin plot of the synthetic data along with the observed data and the validation data.

the validation data set in which the validation performance is quantified using the R^2-score (i.e. see Eq. (9.2)) as the metric. The training times and the R^2-scores for the corresponding ANN are presented in Table 9.11. As seen from the table, the R^2-scores for all the ANNs are above 90 % which indicates that the ANN models are well-validated against the validation data set.

In addition, the training procedure is repeated using the observed data set as the training data set. This is performed on a separate set of ANNs with the same configurations as the set trained with the synthetic data set. The resulting training times and the R^2-scores for the corresponding ANN are presented in Table 9.11. As per the case when trained with the synthetic data set, the R^2-scores for all the ANNs are above 90 % which indicates that the ANN models are well-validated against the validation data set.

It needs to be noted that the R^2-scores are generally lower for the case when the ANN models are trained only with the observed data compared to the case when the ANN models are trained only with the synthetic data. This is due to the former case having a smaller training data set compared to the latter case leading to the ANNs trained using the observed data set having a relatively poorer prediction performance. In addition, it is also observed that the training times of the ANN models are generally longer when the synthetic data is used as the training data due to its relatively large data set compared to the size of the

Model no.	Configuration	Trained with Observed data		Trained with Synthetic data	
		Time $[s]$	R^2-score [%]	Time $[s]$	R^2-score [%]
1	$2-1-1$	0.37	93.76	0.45	98.63
2	$2-2-1$	0.39	92.69	0.78	97.83
3	$2-4-1$	0.34	95.49	0.79	97.16
4	$2-6-1$	0.34	95.17	0.74	97.75
5	$2-8-1$	0.35	95.36	0.75	99.33

Table 9.11: Training times and R^2-score of the respective ANN model for the prediction of the Cantilever beam's free-end displacement D.

observed data set.

9.6.3 Results and discussions

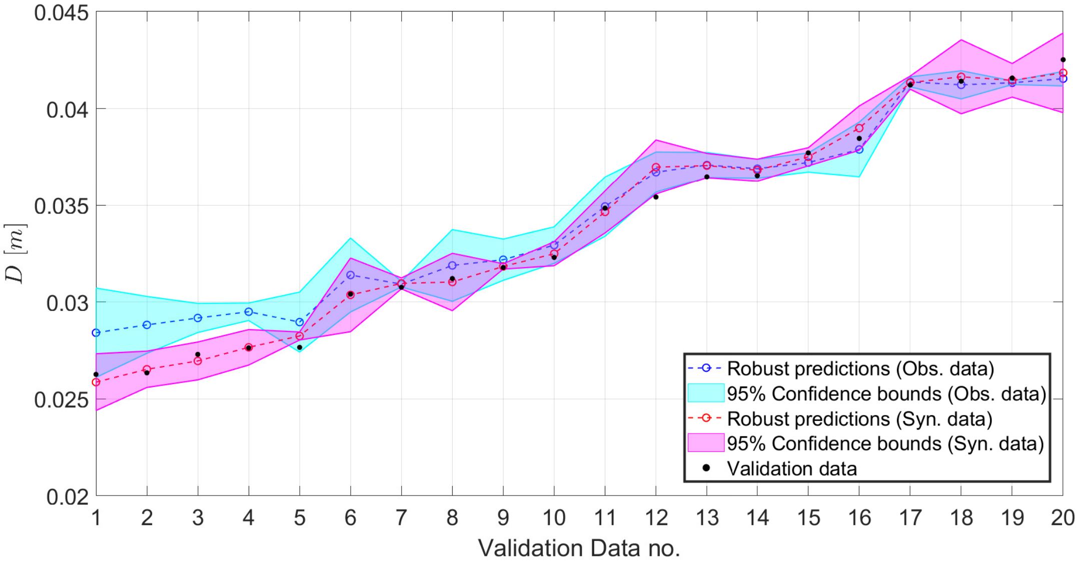


Figure 9.12: Robust predictions, along with its 95 % confidence bounds, of the Cantilever beam's free-end displacement D.

The ABMS method is implemented on the set of ANN models trained with the synthetic data set and the set of ANN models trained with the observed data set. The computation times elapsed by the ABMS algorithm was 7.24 s when implemented on the set of ANN models trained with the synthetic data set, and 5.97 s when implemented on the set of ANN models trained with the observed data set. Detailed descriptions to the ABMS method are presented in Section 9.3.3. The resulting probabilistic prediction of the Cantilever beam's free-end displacement D are presented as graphical plots in Figure 9.12 given both sets of ANN models. From the figure, it can be seen that in general, the robust estimates of D for the case when the synthetic data set is used as the training data set show a higher degree of agreement with the validation data for D compared to the case when the observed data set is used as the training data set. This is especially so for the case of the validation data number 1 to 10 where the robust estimates by the set of ANN models trained using the observed data set show a relatively larger degree of deviation from the validation data values for D.

In addition, it can also be seen that while the 95 % confidence bounds have generally the same width in both cases, such bounds enclose nearly all the validation data values for D for the case of the set of the ANN models trained with the synthetic data set with the exception of data numbers 5 and 12. On the other hand, for the case of the set of the ANN models trained with the observed data set, the 95 % confidence bounds fail to enclose data numbers 1, 2, 3, 4, 5, 12, 19, and 20. These observations highlight a significantly improved probabilistic prediction performance via the ABMS method using the set of ANN models trained using the synthetic data set.

A reason to account for the improved probabilistic prediction performance via the ABMS method using the set of ANN models trained using the synthetic data set is due to the larger data set which is used to train the ANN models leading to a relatively better prediction performance compared to the ANN models trained only with the relatively smaller observed data set. This is supported by the results of the R^2-scores for both sets of ANN models as presented in Table 9.11 to which discussions have been presented in Section 9.6.2.

Finally, the resulting posterior probability $P(M_v, y_v|\boldsymbol{D})$ (i.e. see Eq. (9.3)) associated with each of the ANN models when trained solely with the observation data and those trained solely with the synthetic data are presented graphically in Figure 9.13. It needs to be noted that in both cases, when the ANNs are trained solely with the observe data and those trained solely with the synthetic data, ANN model 5 has the highest value of posterior probability $P(M_v, y_v|\boldsymbol{D})$ associated with it.

9.6.4 Appendix conclusion

The appendix section has presented a numerical example in the form of a Cantilever beam set-up subjected to a $2D$ external force applied to its free-end. The purpose of the example is to provide a comparison in the prediction performance by the ABMS method on a validation

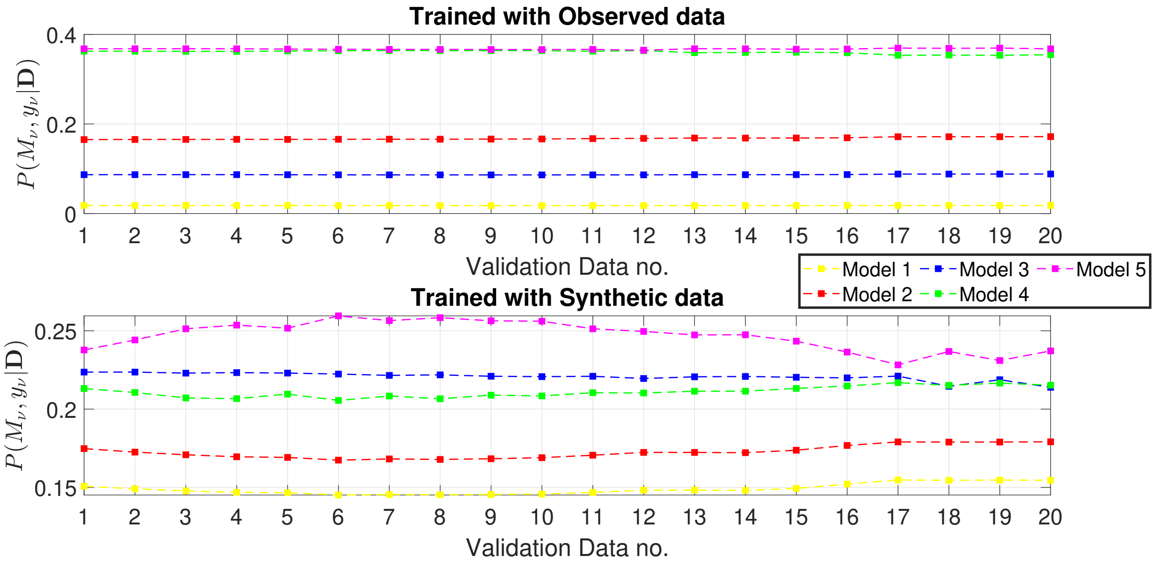


Figure 9.13: Graphical plot of the resulting posterior probability of each of the given ANN model in predicting the corresponding data points associated with the beam's free-end displacement D.

data set given a set of ANN models trained with a synthetic data set generated via a stochastic data enhancement approach proposed in Section 9.3.1 against a a set of ANN models trained only with the observed data set. Results have shown a significantly improved probabilistic prediction performance by the ABMS when using a set of ANN models trained with the synthetic data set. This demonstrates the feasibility of the proposed framework set forth in the chapter.

To allow for the reproducibility the results presented in the section, codes and algorithms used for the study are also made available on GitHub via: `https://github.com/Adolphus8/Project_PROMAP.git`

Chapter 10

Conclusion

Abstract

The chapter will first present a summary of the research contributions as well as the research objectives achieved. From which, recommendations to the future research works, which serve as extensions to the research content presented, will be provided before concluding the chapter and the dissertation.

10.1 Concluding remarks

The research works presented in the dissertation has provided approaches towards improving the robustness and efficiency of sampling tools employed for Bayesian parameter identification, model updating, and model selection in the context of the engineering applications. To demonstrate the robustness and feasibility of the proposed approaches, numerical and experimental case studies were presented and the results were verified against the corresponding reference values. Although the discussions of the developed techniques in the dissertation are limited within the structural and nuclear engineering disciplines, these approaches can still be potentially applied beyond these domains such as finance [328, 329] and climate forecasting [330, 331]. As the focus of the dissertation revolves mainly about the development of numerical and computational tools, the codes to the Transitional Ensemble Markov Chain Monte Carlo sampler, Sequential Ensemble Monte Carlo sampler, and their respective application examples are made available as open-source software for the readers to refer and reproduce the results presented. In summary, the dissertation has achieved the following:

Firstly, a background introduction to the concept of model updating was presented where the description to the deterministic and probabilistic model updating approaches, as well as an evaluation between them, are presented. The Bayesian model updating framework was subsequently identified and set as the context of discussion and basis of the research works presented in the dissertation. The research challenges are identified from which the research objectives are defined. This provided the direction and structure for the dissertation.

Following which, a review on the concept of the Bayesian model updating framework is presented. This included detailed explanation behind the mathematical derivations, the interpretation of the mathematical terms defined in the concept, as well as an extensive literature review on recent developments within that domain. From there, a detailed explanation behind the sampling algorithm is provided for the following Monte Carlo sampler techniques: 1) Markov Chain Monte Carlo; 2) Transitional Markov Chain Monte Carlo; and 3) Sequential Monte Carlo samplers. To illustrate the implementation of the above samplers and evaluate the sampling performance between them, 3 case studies have been presented consisting of 2 numerical examples and an experimental application. The comparisons made between the 3 sampling approaches are done on the basis of their computational efficiency and the precision of their estimates on the inferred parameter(s). From this series of investigations, the strengths and limitations of each sampling approach are presented. This helped provide the motivation behind developing improved sampling algorithms to address some of the limitations which were presented.

With that, the dissertation proceeds to present the proposed Transitional Ensemble Markov Chain Monte Carlo sampler which serves as a new variant to the existing class of Transitional Markov Chain Monte Carlo samplers. To improve its sampling performance and effectiveness from highly-skewed and complex-shaped distribution, the Affine-invariant Ensemble sampler was incorporated as the Markov Chain Monte Carlo kernel along with an improved resampling strategy to update the samples. In addition, an adaptive tuning algorithm was introduced to automatically tune the step-size parameter of the Affine-invariant Ensemble sampler which, in turn, allowed for the acceptance rates of the sampler to be controlled within optimal bounds. In doing so, it makes the proposed sampler free from tuning by the user. To demonstrate its feasibility and verify its results, the proposed sampler has been implemented along with the traditional Transitional Markov Chain Monte Carlo sampler in 2 numerical examples which highlighted 4 key results: 1) the proposed sampler is generally able to yield estimates with a lower standard error than that of the Transitional Markov Chain Monte Carlo sampler; 2) the proposed sampler takes relatively shorter time to generate samples compared to the Transitional Markov Chain Monte Carlo sampler as a result of utilising an efficient sample updating strategy which allows samples to be updated in parallel at an ensemble level via the Affine-invariant Ensemble sampler; 3) the proposed sampler demonstrates an improved mixing performance given that its transition step-size is shown to be generally greater than those obtained by the Transitional Markov Chain Monte Carlo sampler; and 4) the proposed sampler is able to generate samples which explore the sample space better compared to the Transitional Markov Chain Monte Carlo sampler in the case of sampling from a 4-peaked posterior.

To validate the proposed Transitional Ensemble Markov Chain Monte Carlo sampler, 2 experimental problems have been presented involving experimental data and performing Bayesian inference and model updating under model uncertainty. The first of which is the $2DoF$ shear aluminium frame problem where the sampler is implemented to infer the moveable mass positions, used to represent the damage location, on the structure. Unlike previous problems discussed up to this point, the uncertainty over the likelihood functions, introduced apriori in [1], is considered. To provide a robust uncertainty quantification over the inference of the mass positions, the Robust Bayes framework was implemented to account for the uncertainty over the choice of likelihood function (i.e. model uncertainty) and provide a 95 % Credible Interval over the inferred mass positions. The inference results generally agree with the reference values obtained from the experiment campaign. In addition, a discussion was also provided to account for the cases where the 95 % Credible Interval of the inference results do not encompass the true mass positions. Furthermore, a comparison was made between the sampling performance of the Transitional Ensemble Markov Chain Monte Carlo sampler in sampling from each configuration of the posterior. The results showed that the proposed sampler required the highest number of iterations in sampling from the posterior which comprised of the Uniform prior and the Normal distribution likelihood function due to its relatively rapid posterior contraction from the prior.

The second problem involves the analysis of a dynamical black-box system to which it is described by an Uncertainty Model consisting of both aleatory and epistemic model parameters. In the first part of the study, the goal would be to calibrate the Uncertainty Model given a set of response signal data. To do so, the Transitional Ensemble Markov Chain Monte Carlo sampler is implemented to perform Bayesian model updating. For the analysis, 2 Uncertainty Models were proposed, each differing in the aleatory model used for the aleatory model parameters and the likelihood function set-up. For the first Uncertainty Model, the Beta distribution is identified based on the log-evidence results from a family of distributions considered and the model updating was done with the likelihood function incorporating the Wasserstein distance as the stochastic distance metric and a subset of the Fast Fourier Transformed signal output as the data. For the second Uncertainty Model, the Staircase Density Function is considered and the model updating was done with the likelihood function incorporating the Bhattacharyya distance as the stochastic distance metric and the batched time-domain signal output as the data. Results concluded that the second Uncertainty Model yielded much more precise results in its probabilistic model predictions over the response signal output. In the second part of the study, a sensitivity analysis was conducted to rank the sensitivity of the epistemic model parameters. To do so, an adaptive pinching approach is introduced which considers the entire epistemic space defined by the respective Uncertainty Model at a low computational cost.

After which, the conceptual framework behind the design of the Transitional Ensemble Markov Chain Monte Carlo sampler is extended towards the development of the Sequential Ensemble Monte Carlo sampler for the identification of time-varying parameters using Sequential Bayesian inference. This serves as a new variant to the existing class of Sequential Monte Carlo samplers. In addition to the features that have been included and discussed in proposing the Transitional Ensemble Markov Chain Monte Carlo sampler, the Sequential Ensemble Monte Carlo sampler is made more robust with the introduction of "virtual" iterations which serves to not only allow for the automatic tuning of the step-size parameter, but also ensures that the acceptance rate of the sampler converges towards the user-defined target value. To complement the study, an investigation into the existence of inherent bounds on the acceptance rate values is also performed which concluded that the analyst need not be concerned over the possibility of the acceptance rates falling below the optimal bounds and that it is only essential to ensure that such value are not too high (i.e. above 0.8). To validate the Sequential Ensemble Monte Carlo sampler, the proposed sampler is implemented on a single-storey structure subjected to a time-varying Coulomb friction where the objectives are to identify the Coulomb friction, natural frequency of the structure, measurement errors, and the most probable Markov kernel. The results not only highlighted the robustness of the proposed sampler in identifying the most probable Markov model under uncertainty, but are also verified against those obtained by the traditional Sequential Monte Carlo sampler.

Finally, the dissertation proceeds to discuss a novel framework which seeks to merge Artificial Intelligence tools with Bayesian statistics to yield probabilistic predictions over the nuclear material properties under sparse data and model uncertainty. For this study, a particular steel material is considered to which the data associated with the identified input and target features are enhanced through a stochastic approach. Following which, 5 Artificial Neural Network models were constructed and considered for the prediction of each material property to introduce elements of model uncertainty. From there, the Adaptive Bayesian Model Selection method was implemented to introduce Bayesian statistics to the prediction by the set of Artificial Neural Network models and yield robust estimates and its associated confidence bounds. Results have shown that the confidence bounds of the model predictions generally enclose the actual experimental data which verifies the proposed framework.

10.2 Future works

A series of thorough literature reviews and investigations were conducted through the course of this doctoral candidature which resulted in significant scientific contributions towards the discipline of Uncertainty Quantification within the engineering field. While the proposed approaches have demonstrated its robustness and efficiency based on the results presented, these methods have not been proven to be entirely perfect as they are yet to be applied in every possible engineering set-ups. This provided room for further improvements and helped open up 5 follow-up research works which will serve as the research plan for the future. They seek to investigate the following:

1) **Defining the target acceptance rate of the Affine-invariant Ensemble sampler:**

   In Chapter 4, an adaptive-tuning algorithm is proposed which seeks to adaptively tune the step-size hyper-parameter of the Affine-invariant Ensemble sampler in the Markov Chain Monte Carlo step. Such tuning is based on a reference target acceptance rate defined in [150] which is based on the study of the Metropolis-Hastings Markov Chain Monte Carlo sampler. In addition, the optimal acceptance rates bounds of $[0.15, 0.50]$ was also obtained from [150]. Currently, there is no literature that looks into the mathematical derivation of the target acceptance rate and its optimal bounds in the case of the Affine-invariant Ensemble sampler. A future research work seeks to investigate this to contribute towards the theoretical findings behind the sampler's performances.

2) **Uncertainty Model analysis for the NASA-Langley Challenge problem:**

   In Chapter 6, 2 Uncertainty Models were proposed to quantify the uncertain model parameters of the dynamical back-box system. Although it was found that the set-up for the second Uncertainty Model is less uncertain, it was not investigated as to what are the factors that allowed this to be so. For instance, did the choice of the Staircase Density Function used to model the distribution function of the aleatory model parameters play a significant role due to its flexibility in modelling complex or unknown distributions? Did the choice of the stochastic distance metric in the likelihood function affected the results over the identification of the distribution model of the aleatory model parameters? Did the choice of the data type have an impact on the resulting Uncertainty Model calibration results? To address this, a further 4 Uncertainty Models can be set-up and calibrated via Bayesian model updating to which the analysis can be done as presented in Chapter 6. These Uncertainty Models and their configurations are summarised in Table 10.1. While the preliminary studies and results are presented in a paper accepted in the Journal of Physics: Conference series, further investigations shall look into the reliability analysis of the given design point for each Uncertainty Model. This seeks to compare the imprecise failure probability estimates by each Uncertainty Model and provide an understanding as to how the different configurations of the Uncertainty Model would affect the uncertainty of such estimates.

Uncertainty Model	Distribution type	Data type	Distance metric
3	Beta	Time-domain	Bhattacharyya
4	Beta	Frequency-domain	Bhattacharyya
5	Beta	Time-domain	Wasserstein
6	Staircase Density Function	Frequency-domain	Wasserstein

Table 10.1: Set-up of the additional Uncertainty Models for subsequent investigations.

3) **An inverse approach towards interval analysis under sparse data and model uncertainty:**

   In Chapter 9, the gaps in the experimental data are filled through a stochastic data-enhancement method. For this extended investigation, the gaps in the data would be represented as intervals. In doing so, this would allow for model predictions to be represented as intervals given a set of interval model inputs. An inverse problem will be presented as follows: Given a user-defined tolerated level of confidence interval on the model prediction of the steel material property, what would be the corresponding interval of the input data that would yield such interval output? Such problem can be addressed using Bayesian model updating approach to yield interval estimates of the input parameter(s) via the improved Transitional Ensemble Markov Chain Monte Carlo sampler from the first extended research work. In addition, a set of Artificial Neural Network models can be considered to introduce model uncertainty and the most probable model can be identified using the evidence term computed from the aforementioned sampler.

4) **Identifying Markov kernel parameter(s) under uncertainty using the Sequential Ensemble Monte Carlo sampler:**

   In Chapter 8, the Sequential Ensemble Monte Carlo sampler is implemented on a single-story structure subjected to a time-varying Coulomb friction and considers the case where the model parameter(s) of the Markov kernel are fixed and known. However, in general, such parameter(s) are not known and would need to be inferred. Hence, this presents a need to further investigate the robustness of the proposed algorithm in inferring the model parameters of the Markov kernel in addition to the time-varying parameter(s) that is/are being estimated.

5) **Consideration of correlated and time-varying measurement errors:**

   The numerical and experimental case studies presented in the dissertation assumes that the measurement errors are time-invariant and that they are independent between each measurement. As such, future investigations will look towards problems where such errors vary with time where its identification will be performed using the proposed Sequential Ensemble Monte Carlo sampler and a time-varying likelihood function. Such has been highlighted in Chapter 3 as an open research topic. In addition, to introduce further complexities and broaden the generality of the applications, correlations will be considered as an extension to this study to investigate the robustness of the sampling performance and identification of the time-varying parameter(s) by the proposed sampler under challenging conditions. The purpose of this is to validate the sampler and demonstrate its feasibility to analyse real-world engineering problems.

Bibliography

[1] R. Rocchetta, M. Broggi, Q. Huchet, and E. Patelli, "On-line Bayesian Model Updating for Structural Health Monitoring," *Mechanical Systems and Signal Processing*, vol. 103, pp. 174–195, 2018.

[2] Y. Govers, H. H. Khodaparast, M. Link, and J. Mottershead, "A Comparison of Two Stochastic Model Updating Methods using the DLR-AIRMOD Test Structure," *Mechanical Systems and Signal Processing*, vol. 52-53, pp. 105–114, 2015.

[3] E. Patelli, Y. Govers, M. Broggi, H. M. Gomes, M. Link, and J. E. Mottershead, "Sensitivity or Bayesian model updating: A comparison of techniques using the DLR-AIRMOD test data," *Archive of Applied Mechanics*, vol. 87, pp. 905–925, 2017.

[4] P. Liang, J. E. Mottershead, and F. A. DiazDelaO, "Model Updating with the Kriging Predictor: Effect of Code Uncertainty," *Proceedings of the International Conference on Noise and Vibration Engineering 2016 (ISMA 2016) and International Conference on Uncertainty in Structural Dynamics (USD 2016), Leuven, Belgium*, vol. 1, pp. 4363–4375, 2016.

[5] J. Goodman and J. Weare, "Ensemble samplers with affine invariance," *Communications in Applied Mathematics and Computational Science*, vol. 5, pp. 65–80, 2010.

[6] S. T. Thorton and J. B. Marion, *Classical Dynamics of Particles and Systems.* Brooks/Cole, 5 ed., 2004. ISBN: 978-0534408961.

[7] D. Himmelblau, *Applied Nonlinear Programming.* McGraw-Hill, 1972. ISBN: 0070289212.

[8] H. H. Khodaparast, J. E. Mottershead, and K. J. Badcock, "Interval model updating with irreducible uncertainty using the Kriging predictor," *Mechanical Systems and Signal Processing*, vol. 25, pp. 1204–1226, 2011.

[9] L. Marino and A. Cicirello, "Experimental investigation of a single-degree-of-freedom system with Coulomb friction," *Nonlinear Dynamics*, vol. 99, p. 1781–1799, 2020.

[10] K. Worden, *Nonlinearity in Structural Dynamics: Detection, Identification and Modelling*. CRC Press, 2019.

[11] M. I. Friswell and J. E. Mottershead, *Finite Element Model Updating in Structural Dynamics*. Dordrecht: Springer, 2011. ISBN: 978-9048145355.

[12] H. Waisman, E. Chatzi, and A. W. Smyth, "Detection and quantification of flaws in structures by the extended finite element method and genetic algorithms," *International Journal for Numerical Methods in Engineering*, vol. 82, pp. 303–328, 2009.

[13] E. Simoen, G. D. Roeck, and G. Lombaert, "Dealing with uncertainty in model updating for damage assessment: A review," *Mechanical Systems and Signal Processing*, vol. 56-57, pp. 123–149, 2015.

[14] I. Hlaváěek, J. Chleboun, and I. Babuška, "Reality, Mathematics, and Computation," *North-Holland Series in Applied Mathematics and Mechanics Uncertain Input Data Problems and the Worst Scenario Method*, vol. , pp. 1–49, 2004.

[15] J. Helton, "An exploration of alternative approaches to the representation of uncertainty in model predictions," *Reliability Engineering and System Safety*, vol. 85, pp. 39–71, 2004.

[16] J. Helton, "Uncertainty and sensitivity analysis in the presence of stochastic and subjective uncertainty," *Journal of Statistical Computation and Simulation*, vol. 57, pp. 3–76, 1997.

[17] R. L. Winkler, "Uncertainty in probabilistic risk assessment," *Reliability Engineering and System Safety*, vol. 54, pp. 127–132, 1996.

[18] Z. Meng, Y. Pang, Y. Pu, and X. Wang, "New hybrid reliability-based topology optimization method combining fuzzy and probabilistic models for handling epistemic and aleatory uncertainties," *Computer Methods in Applied Mechanics and Engineering*, vol. 363, p. 112886, 2020.

[19] Y. Luo, A. Li, and Z. Kang, "Reliability-based design optimization of adhesive bonded steel–concrete composite beams with probabilistic and non-probabilistic uncertainties," *Engineering Structures*, vol. 33, pp. 2110–2119, 2011.

[20] B. Ni, C. Jiang, and X. Han, “An improved multidimensional parallelepiped non-probabilistic model for structural uncertainty analysis,” *Applied Mathematical Modelling*, vol. 40, pp. 4727–4745, 2016.

[21] B. Ni, C. Jiang, and Z. Huang, “Discussions on non-probabilistic convex modelling for uncertain problems,” *Applied Mathematical Modelling*, vol. 59, pp. 54–85, 2018.

[22] M. Rashki, “Hybrid control variates-based simulation method for structural reliability analysis of some problems with low failure probability,” *Applied Mathematical Modelling*, vol. 60, pp. 220–234, 2018.

[23] H. H. Khodaparast, J. E. Mottershead, and K. J. Badcock, “Interval Model Updating with Irreducible Uncertainty using the Kriging Predictor,” *Mechanical Systems and Signal Processing*, vol. 25, p. 1204–1226, 2011.

[24] C. J. Roy and W. L. Oberkampf, “A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing,” *Computer Methods in Applied Mechanics and Engineering*, vol. 200, pp. 2131–2144, 2011.

[25] W. L. Oberkampf, J. C. Helton, C. A. Joslyn, S. F. Wojtkiewicz, and S. Ferson, “Challenge problems: Uncertainty in system response given uncertain parameters,” *Reliability Engineering and System Safety*, vol. 85, pp. 11–19, 2004.

[26] L. G. Crespo and S. P. Kenny, “The NASA Langley UQ Challenge on Optimization Under Uncertainty,” *Mechanical Systems and Signal Processing*, vol. 152, p. 107405, 2021.

[27] A. D. Kiureghian and O. Ditlevsen, “Aleatory or epistemic? Does it matter?,” *Structural Safety*, vol. 31, pp. 105–112, 2009.

[28] R. Rocchetta, M. Broggi, and E. Patelli, “Do we have enough data? Robust reliability via uncertainty quantification,” *Applied Mathematical Modelling*, vol. 54, pp. 710–721, 2018.

[29] M. Faes and D. Moens, “Recent trends in the modeling and quantification of non-probabilistic uncertainty,” *Archives of Computational Methods in Engineering*, vol. 27, p. 633–671, 2019.

[30] J. Wolberg, *Data Analysis Using the Method of Least Squares Extracting the Most Information from Experiments*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. ISBN: 978-3540317203.

[31] H. F. Lam, J. Hu, F. L. Zhang, and Y. C. Ni, "Markov chain Monte Carlo-based Bayesian model updating of a sailboat-shaped building using a parallel technique," *Engineering Structures*, vol. 193, pp. 12–27, 2019.

[32] M. Aghagholizadeh and F. N. Catbas, "A Review of Model Updating Methods for Civil Infrastructure Systems," *Computational Techniques for Civil and Structural Engineering*, 2015.

[33] M. Kitahara, S. F. Bi, M. Broggi, and M. Beer, "Nonparametric Bayesian stochastic model updating with hybrid uncertainties," *Mechanical Systems and Signal Processing*, vol. 163, p. 108195, 2022.

[34] S. Ferson, V. Kreinovick, L. Ginzburg, and F. Sentz, "Constructing Probability Boxes and Dempster-Shafer Structures," *Sandia National Laboratories*, vol. 4015, p. , 2002.

[35] K. Sentz and S. Ferson, "Combination of Evidence in Dempster-Shafer Theory," *Sandia National Laboratories*, 2002.

[36] M. Beer, S. Ferson, and V. Kreinovich, "Imprecise probabilities in engineering analyses," *Mechanical Systems and Signal Processing*, vol. 37, pp. 4–29, 2013.

[37] J. L. Beck and L. S. Katafygiotis, "Updating Models and Their Uncertainties. I: Bayesian Statistical Framework," *Journal of Engineering Mechanics*, vol. 124, pp. 455–461, 1998.

[38] L. S. Katafygiotis and J. L. Beck, "Updating Models and Their Uncertainties. II: Model Identifiability," *Journal of Engineering Mechanics*, vol. 124, p. 463–467, 1998.

[39] L. S. Katafygiotis, C. Papadimitriou, and H. Lam, "A probabilistic approach to structural model updating," *Soil Dynamics and Earthquake Engineering*, vol. 17, pp. 495–507, 1998.

[40] H. F. Lam, L. S. Katafygiotis, and N. C. Mickleborough, "Application of a Statistical Model Updating Approach on Phase I of the IASC-ASCE Structural Health Monitoring Benchmark Study," *Journal of Engineering Mechanics*, vol. 130, pp. 34–48, 2004.

[41] S. K. Au, “Connecting Bayesian and frequentist quantification of parameter uncertainty in system identification,” *Mechanical Systems and Signal Processing*, vol. 29, pp. 328–342, 2012.

[42] D. Giagopoulos, A. Arailopoulos, V. Dertimanis, C. Papadimitriou, E. Chatzi, and K. Grompanopoulos, “Structural health monitoring and fatigue damage estimation using vibration measurements and finite element model updating,” *Structural Health Monitoring*, vol. 18, pp. 1189–1206, 2019.

[43] J. Ching and J. L. Beck, “New Bayesian Model Updating Algorithm Applied to a Structural Health Monitoring Benchmark,” *Structural Health Monitoring: An International Journal*, vol. 3, pp. 313–332, 2004.

[44] K. V. Yuen, J. Beck, and L. S. Katafygiotis, “Efficient model updating and health monitoring methodology using incomplete modal data without mode matching,” *Structural Control Health Monitoring*, vol. 13, pp. 91–107, 2006.

[45] S. E. Azam, C. Papadimitriou, and E. Chatzi, “Recursive Bayesian filtering for displacement estimation via output-only vibration measurements,” *Proceedings of the 2014 World Congress on Advances in Civil, Environmental, and Materials Research*, vol. , p. , 2014.

[46] A. Lye, A. Cicirello, and E. Patelli, “Sampling methods for solving Bayesian model updating problems: A tutorial,” *Mechanical Systems and Signal Processing*, vol. 159, p. 107760, 2021.

[47] J. S. Liu, F. Liang, and W. H. Wong, “The Multiple-Try Method and Local Optimization in Metropolis Sampling,” *Journal of the American Statistical Association*, vol. 95, p. 121–134, 2000.

[48] J. Besag, “Comments on “Representations of Knowledge in Complex Systems” by U. Grenander and M. I. miller,” *Journal of the Royal Statistical Society, Series B*, vol. 56, p. 591–592, 1994.

[49] S. Duane, A. Kennedy, B. J. Pendleton, and D. Roweth, “Hybrid Monte Carlo,” *Physics Letters B*, vol. 195, p. 216–222, 1987.

[50] W. K. Hastings, “Monte Carlo Sampling Methods using Markov Chains and their Applications,” *Biometrika*, vol. 57, p. 97–109, 1970.

[51] J. Y. Ching and Y. C. Chen, "Transitional Markov Chain Monte Carlo Method for Bayesian Model Updating, Model Class Selection, and Model Averaging," *Journal of Engineering Mechanics*, vol. 133, no. 7, 2007.

[52] P. D. Moral, A. Doucet, and A. Jasra, "Sequential Monte Carlo Samplers," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 68, pp. 411–436, 2006.

[53] D. J. Jerez, A. H. Jensen, and M. Beer, "An effective implementation of reliability methods for Bayesian model updating of structural dynamic models with multiple uncertain parameters," *Reliability Engineering and System Safety*, vol. 225, p. 108634, 2022.

[54] ANSIC, "Digital technologies to support development, deployment, operation and decommissioning of advanced nuclear technologies," 2021.

[55] J. P. Noel and M. Schoukens, "Cross-fertilising research in nonlinear system identification between the mechanical, control and machine learning fields: Editorial statement," *Mechanical Systems and Signal Processing*, vol. 130, p. 2019, 213-220.

[56] M. C. Kennedy and A. O'Hagan, "Bayesian calibration of Computer Models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 3, p. 63, 2001.

[57] G. Capellari, E. Chatzi, and S. Mariani, "Optimal sensor placement through bayesian experimental design: Effect of measurement noise and number of sensors," *Proceedings*, vol. 1, p. 41, 2016.

[58] P. G. Daneshmand and R. Jafari, "A 3D hybrid BE–FE solution to the forward problem of electrical impedance tomography," *Engineering Analysis with Boundary Elements*, vol. 37, pp. 757–764, 2013.

[59] Z. Xu, Q. Li, and W. He, "Analytical Solution for the Forward Problem of Magnetic Induction Tomography with Multi-layer Sphere Model," *Lecture Notes in Computer Science Life System Modeling and Intelligent Computing*, vol. , pp. 42–50, 2010.

[60] Y. Xu and F. Dong, "Galerkin boundary element method for the forward problem of ERT," *Flow Measurement and Instrumentation*, vol. 21, pp. 172–177, 2010.

[61] T. Shi, J. Ma, H. Huang, Y. Qiu, H. Zeng, Z. Li, and D. Qian, "Variance reduction for global response problem based on forward Monte Carlo calculation," *Nuclear Engineering and Design*, vol. 322, pp. 291–300, 2017.

[62] A. Elidrysy, S. Harir, A. Zouhair, and Y. Boughaleb, "Simulation of an extended 3D mixed Ising model by Monte Carlo method," *Materials Today: Proceedings*, vol. 30, pp. 993–997, 2020.

[63] S. Li and L. Caracoglia, "Surrogate Model Monte Carlo simulation for stochastic flutter analysis of wind turbine blades," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 188, pp. 43–60, 2019.

[64] G. Dahlquist, B. Ake, and N. Anderson, *Numerical Methods.* Mineola (New York): Dover, 2003. ISBN: 978-0486428079.

[65] J. H. Albert, "Teaching Bayesian Statistics Using Sampling Methods and MINITAB," *The American Statistician*, vol. 47, pp. 182–191, 1993.

[66] C. E. Papadopoulos and H. Yeung, "Uncertainty Estimation and Monte Carlo Simulation Method," *Flow Measurement and Instrumentation*, vol. 12, pp. 291–298, 2001.

[67] N. T. Thomopoulos, *Essentials of Monte Carlo Simulation.* Springer, 2015. ISBN: 978-1461460213.

[68] R. J. Rossi, *Mathematical statistics: An introduction to likelihood based inference.* Hoboken, NJ: John Wiley and Sons, 2018. ISBN: 978-1118771044.

[69] H. Karimi and K. B. Mcauley, "A maximum-likelihood method for estimating parameters, stochastic disturbance intensities and measurement noise variances in nonlinear dynamic models with process disturbances," *Computers and Chemical Engineering*, vol. 67, pp. 178–198, 2014.

[70] Z. Szabo, X. Liu, and L. Xiang, "Semiparametric sieve maximum likelihood estimation for accelerated hazards model with interval-censored data," *Journal of Statistical Planning and Inference*, vol. 205, pp. 175–192, 2020.

[71] F. Li, K. Li, K. Lu, and Z. Li, "Random noise suppression and parameter estimation for Magnetic Resonance Sounding signal based on maximum likelihood estimation," *Journal of Applied Geophysics*, vol. 176, p. 104007, 2020.

[72] K. V. Yuen, *Bayesian Methods for Structural Dynamics and Civil Engineering.* Singapore: John Wiley and Sons Asia, 2010. ISBN: 978-0470824559.

[73] T. Bayes and Price, “LII. An Essay towards Solving a problem in the Doctrine of Chances. By the late Rev. Mr. Bayes, F. R. S. communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S.,” *Philosophical Transactions of the Royal Society of London*, vol. 53, pp. 370–418, 1763.

[74] M. Faes, M. Broggi, E. Patelli, Y. Govers, J. Mottershead, M. Beer, and D. Moens, “A multivariate interval approach for inverse uncertainty quantification with limited experimental data,” *Mechanical Systems and Signal Processing*, vol. 118, pp. 534–548, 2019.

[75] E. T. Jaynes, *Information theory and statistical mechanics.* Stanford, CA: Microwave Laboratory, 1957. W. W. Hansen Laboratories of Physics, Stanford University.

[76] E. T. Jaynes, “The Relation of Bayesian and Maximum Entropy Methods,” *Maximum-Entropy and Bayesian Methods in Science and Engineering*, vol. , p. 25–29, 1988.

[77] R. D. Teloli, S. D. Silva, T. G. Ritto, and G. Chevallier, “Bayesian model identification of higher-order frequency response functions for structures assembled by bolted joints,” *Mechanical Systems and Signal Processing*, vol. 151, p. 107333, 2021.

[78] W. J. Yan, D. Chronopoulos, S. C. Chinchilla, K. V. Yuen, and C. Papadimitriou, “A fast bayesian inference scheme for identification of local structural properties of layered composites based on wave and finite element-assisted metamodeling strategy and ultrasound measurements,” *Mechanical Systems and Signal Processing*, vol. 143, p. 106802, 2020.

[79] S. He and C. T. Ng, “Guided wave-based identification of multiple cracks in beams using a bayesian approach,” *Mechanical Systems and Signal Processing*, vol. 84, pp. 324–345, 2017.

[80] E. Patelli, M. Broggi, Y. Govers, and J. E. Mottershead, “Model Updating Strategy of the DLR-AIRMOD Test Structure,” *Procedia Engineering*, vol. 199, pp. 978–983, 2017.

[81] T. G. Ritto, R. Sampaio, and R. R. Aguiar, “Uncertain Boundary Condition Bayesian Identification from Experimental Data: A Case Study on a Cantilever Beam,” *Mechanical Systems and Signal Processing*, vol. 68-69, pp. 176–188, 2016.

[82] P. L. Green and S. Maskell, “Estimating the Parameters of Dynamical Systems from Big Data using Sequential Monte Carlo Samplers,” *Mechanical Systems and Signal Processing*, vol. 93, pp. 379–396, 2018.

[83] H. Jalali, H. H. Khodaparast, H. Madinei, and M. Friswell, “Stochastic modelling and updating of a joint contact interface,” *Mechanical Systems and Signal Processing*, vol. 129, pp. 645–658, 2019.

[84] T. Rogers, K. Worden, and E. Cross, “On the application of Gaussian process latent force models for joint input-state-parameter estimation: With a view to Bayesian operational identification,” *Mechanical Systems and Signal Processing*, vol. 140, p. 106580, 2020.

[85] E. Jennings and M. Madigan, “astroABC: An Approximate Bayesian Computation Sequential Monte Carlo sampler for cosmological parameter estimation,” *Astronomy and Computing*, vol. 19, pp. 16–22, 2017.

[86] Y. Wang, Y. Ni, and X. Wang, “Real-time defect detection of high-speed train wheels by using Bayesian forecasting and dynamic model,” *Mechanical Systems and Signal Processing*, vol. 139, p. 106654, 2020.

[87] H. Rappel, L. Beex, L. Noels, and S. Bordas, “Identifying elastoplastic parameters with Bayes’ theorem considering output error, input error and model uncertainty,” *Probabilistic Engineering Mechanics*, vol. 55, pp. 28–41, 2019.

[88] B. Goller and G. Schuëller, “Investigation of model uncertainties in Bayesian structural model updating,” *Journal of Sound and Vibration*, vol. 330, pp. 6122–6136, 2011.

[89] B. Han, T. Y. Xiang, and H. B. Xie, “A Bayesian inference framework for predicting the long-term deflection of concrete structures caused by creep and shrinkage,” *Engineering Structures*, vol. 142, p. 46–55, 2017.

[90] Z. P. Bazant and J. C. Chern, “Bayesian Statistical Prediction of Concrete Creep and Shrinkage,” *ACI Journal Proceedings*, vol. 81, pp. 319–330, 1984.

[91] H. Keitel, A. Dimmig-Osburg, L. Vandewalle, and L. Schueremans, “Selecting creep models using Bayesian methods,” *Materials and Structures*, vol. 45, pp. 1513–1533, 2012.

[92] X. Guan, R. Jha, and Y. Liu, “Model selection, updating, and averaging for probabilistic fatigue damage prognosis,” *Structural Safety*, vol. 33, pp. 242–249, 2011.

[93] Q. Huang, P. Gardoni, and S. Hurlebaus, “A probabilistic damage detection approach using vibration-based nondestructive testing,” *Structural Safety*, vol. 38, pp. 11–21, 2012.

[94] W. J. Yan and L. S. Katafygiotis, “A novel Bayesian approach for structural model updating utilizing statistical modal information from multiple setups,” *Structural Safety*, vol. 52, pp. 260–271, 2015.

[95] P. Peralta, R. O. Ruiz, and A. A. Taflanidis, “Bayesian identification of electromechanical properties in piezoelectric energy harvesters,” *Mechanical Systems and Signal Processing*, vol. 141, p. 106506, 2020.

[96] L. Lerimonti, I. Venanzi, N. Cavalagli, F. Comodini, and F. Ubertini, “An innovative continuous bayesian model updating method for base-isolated RC buildings using vibration monitoring data,” *Mechanical Systems and Signal Processing*, vol. 139, p. 106600, 2020.

[97] S. H. Cheung and S. Bansal, “A new Gibbs sampling based algorithm for Bayesian model updating with incomplete complex modal data,” *Mechanical Systems and Signal Processing*, vol. 92, pp. 156–172, 2017.

[98] T. Yin and H. Zhu, “Selection of masters in dynamic reduction-based structural health monitoring using Bayesian experimental design,” *Mechanical Systems and Signal Processing*, vol. 150, p. 107294, 2021.

[99] P. Gardner, C. Lord, and R. J. Barthorpe, “Bayesian history matching for structural dynamics applications,” *Mechanical Systems and Signal Processing*, vol. 143, p. 106828, 2020.

[100] Z. Ding, J. Li, and H. Hao, “Structural damage identification using improved Jaya algorithm based on sparse regularization and Bayesian inference,” *Mechanical Systems and Signal Processing*, vol. 132, pp. 211–231, 2019.

[101] A. Lye, A. Cicirello, and E. Patelli, "An efficient and robust sampler for Bayesian inference: Transitional Ensemble Markov Chain Monte Carlo," *Mechanical Systems and Signal Processing*, vol. 167, p. 108471, 2022.

[102] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian Data Analysis*. CRC Press, 3 ed., 2013. ISBN: 978-1439898208.

[103] B. C. Bates and E. P. Campbell, "A markov chain monte carlo scheme for parameter estimation and inference in conceptual rainfall-runoff modeling," *Water Resources Research*, vol. 37, p. 937–947, 2001.

[104] X. Y. Jia and C. Papadimitriou, "Data Features-based Likelihood-informed Bayesian Finite Element Model Updating," *In Proceedings of 3rd International Conference on Uncertainty Quantification in Computational Sciences and Engineering*, vol. 1, pp. 103–113, 2019.

[105] X. Zhou, A. Montazeri, and J. D. Albertson, "Mobile sensing of point-source gas emissions using bayesian inference: An empirical examination of the likelihood function," *Atmospheric Environment*, vol. 218, p. 116981, 2019.

[106] E. Yee, P. R. Kosteniuk, G. M. Chandler, C. A. Biltoft, and J. F. Bowers, "Statistical characteristics of concentration fluctuations in dispersing plumes in the atmospheric surface layer," *Boundary-Layer Meteorology*, vol. 65, p. 69–109, 1993.

[107] I. Senocak, N. W. Hengartner, M. B. Short, and W. B. Daniel, "Stochastic event reconstruction of atmospheric contaminant dispersion using Bayesian inference," *Atmospheric Environment*, vol. 42, pp. 7718–7727, 2008.

[108] D. Wade and I. Senocak, "Stochastic reconstruction of multiple source atmospheric contaminant dispersion events," *Atmospheric Environment*, vol. 74, pp. 45–51, 2013.

[109] S. Bi, M. Broggi, and M. Beer, "The role of the Bhattacharyya distance in stochastic model updating," *Mechanical Systems and Signal Processing*, vol. 117, pp. 437–452, 2019.

[110] A. B. Abdessalem, N. Dervilis, D. Wagg, and K. Worden, "Model selection and parameter estimation in structural dynamics using approximate Bayesian computation," *Mechanical Systems and Signal Processing*, vol. 99, pp. 306–325, 2018.

[111] S. Fang, S. Chen, Y. Lin, and Z. Dong, "Probabilistic damage identification incorporating approximate Bayesian computation with stochastic response surface," *Mechanical Systems and Signal Processing*, vol. 128, pp. 229–243, 2019.

[112] C. Scheidt and J. Caers, "Uncertainty Quantification in Reservoir Performance Using Distances and Kernel Methods – Application to a West Africa Deepwater Turbidite Reservoir," *SPE Journal*, vol. 14, pp. 680–692, 2009.

[113] A. Sankhya, "Reprint of: Mahalanobis, P. C. (1936) "On the Generalised Distance in Statistics"," *Sankhya A*, vol. 80, pp. 1–7, 2018.

[114] A. Bhattacharyya, "On a measure of divergence between two multinomial populations," *The Indian Journal of Statistics*, vol. 7, pp. 401–406, 1946.

[115] H. Raiffa and R. Schlaifer, *Applied Statistical Decision Theory.* Wiley, 2000. ISBN: 978-0471383499.

[116] P. Diaconis and D. Ylvisaker, "Conjugate Priors for Exponential Families," *The Annals of Statistics*, vol. 7, p. 269–281, 1979.

[117] L. D. Brown, *Fundamentals of Statistical Exponential Families: With Applications in Statistical Decision Theory.* Institute of Mathematical Statistics, 1986. ISBN: 978-0940600102.

[118] G. Casella and R. L. Berger, *Statistical Inference.* Brooks/Cole Publishing Company, 1990. ISBN: 978-0534119584.

[119] T. Baldacchino, K. Worden, and J. Rowson, "Robust nonlinear system identification: Bayesian mixture of experts using the t-distribution," *Mechanical Systems and Signal Processing*, vol. 85, pp. 977–992, 2017.

[120] M. Pharr, W. Jakob, and G. Humphreys, "Monte Carlo Integration," *Physically Based Rendering*, vol. , pp. 747–802, 2017.

[121] R. L. Berger, "Estimation: Point and interval," *International Encyclopedia of the Social and Behavioral Sciences*, vol. , pp. 16–20, 2015.

[122] P. L. Green and K. Worden, "Bayesian and Markov chain Monte Carlo methods for identifying nonlinear systems in the presence of uncertainty," *Philosophical Transactions of the Royal Society A.*, vol. 373, p. 20140405, 2015.

[123] P. E. Hadjidoukas, P. Angelikopoulos, D. Rossinelli, D. Alexeev, C. Papadimitriou, and P. Koumoutsakos, “Bayesian Uncertainty Quantification and propagation for Discrete Element Simulations of Granular Materials,” *Computer Methods in Applied Mechanics and Engineering*, vol. 282, p. 218–238, 2014.

[124] R. Rastogi, S. Ghosh, A. K. Ghosh, K. K. Vaze, and P. K. Singh, “Fatigue Crack Growth Prediction in Nuclear Piping using Markov Chain Monte Carlo Simulation,” *Fatigue and Fracture of Engineering Materials and Structures*, vol. 40, p. 145–156, 2016.

[125] S. Rouchier, M. J. Jimenez, and S. Castano, “Sequential Monte Carlo for On-line Parameter Estimation of a Lumped Building Energy Model,” *Energy and Buildings*, vol. 187, pp. 86–94, 2019.

[126] H. Jensen and D. Jerez, “A Bayesian model updating approach for detection-related problems in water distribution networks,” *Reliability Engineering and System Safety*, vol. 185, pp. 100–112, 2019.

[127] D. Straub and I. Papaioannou, “Bayesian Updating with Structural Reliability Methods,” *Journal of Engineering Mechanics*, vol. 141, p. , 2014.

[128] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of State Calculations by Fast Computing Machines,” *The Journal of Chemical Physics*, vol. 21, pp. 1087–1092, 1953.

[129] P. A. Gagniuc, *Markov Chains: From Theory to Implementation and Experimentation.* Chichester, West Sussex: Wiley Blackwell, 2017. ISBN: 978-1119387558.

[130] R. Serfozo, *Basics of Applied Stochastic Processes.* Berlin: Springer Berlin, 2014. ISBN: 978-3540893318.

[131] W. L. Dunn and J. K. Shultis, *Exploring Monte Carlo Methods.* Elsevier, 2011. ISBN: 978-0444515759.

[132] J. K. Kruschke, “Markov Chain Monte Carlo,” *Doing Bayesian Data Analysis*, vol. , pp. 143–191, 2015.

[133] A. E. Gelfand and A. F. M. Smith, “Sampling-Based Approaches to Calculating Marginal Densities,” *Journal of the American Statistical Association*, vol. 85, pp. 398–409, 1990.

[134] W. R. Gilks and P. Wild, “Adaptive Rejection Sampling for Gibbs Sampling,” *Applied Statistics*, vol. 41, pp. 337–348, 1992.

[135] S. Theodoridis, “Monte Carlo Methods,” *Machine Learning*, vol. , pp. 707–744, 2015.

[136] S. Chib, “Markov Chain Monte Carlo Methods: Computation and Inference,” *Handbook of Econometrics*, vol. 5, pp. 3569–3649, 2001.

[137] X. S. Yang, *Introduction to Algorithms for Data Mining and Machine Learning.* Academic Press, 1 ed., 2019. ISBN: 978-0128172162.

[138] S. Brooks, A. Gelman, G. Jones, and X. L. Meng, *Handbook of Markov Chain Monte Carlo.* CRC Press, 2011. ISBN: 978-1420079425.

[139] D. J. Wilkinson, “Parallel Bayesian Computation,” *Handbook of Parallel Computing and Statistics*, pp. 481–512, 2005.

[140] A. E. Brockwell, “Parallel Markov Chain Monte Carlo Simulation by Pre-Fetching,” *Journal of Computational and Graphical Statistics*, vol. 15, pp. 246–261, 2006.

[141] R. M. Neal, “Slice Sampling,” *The Annals of Statistics*, vol. 31, pp. 705–767, 2003.

[142] M. Girolami and B. Calderhead, “Riemann manifold Langevin and Hamiltonian Monte Carlo methods,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 73, pp. 123–214, 2011.

[143] G. O. Roberts and R. L. Tweedie, “Exponential Convergence of Langevin Distributions and Their Discrete Approximations,” *Bernoulli*, vol. 2, p. 341, 1996.

[144] G. O. Roberts and J. S. Rosenthal, “Optimal scaling of discrete approximations to Langevin diffusions,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 60, p. 255–268, 1998.

[145] J. S. Liu, F. Liang, and W. H. Wong, “The Multiple-Try Method and Local Optimization in Metropolis Sampling,” *Journal of the American Statistical Association*, vol. 95, pp. 121–134, 2000.

[146] L. Martino, “A review of multiple try MCMC algorithms for signal processing,” *Digital Signal Processing*, vol. 75, p. 134–152, 2018.

[147] P. J. Green, "Reversible Jump Markov Chain Monte Carlo computation and Bayesian model determination," *Biometrika*, vol. 82, pp. 711–732, 1995.

[148] C. Andrieu and G. O. Roberts, "The pseudo-marginal approach for efficient Monte Carlo computations," *The Annals of Statistics*, vol. 37, pp. 697–725, 2009.

[149] G. O. Roberts and J. S. Rosenthal, "Optimal Scaling for various Metropolis-Hastings Algorithms," *Statistical Science*, vol. 16, pp. 351–367, 2001.

[150] G. O. Roberts, A. Gelman, and W. R. Gilks, "Weak Convergence and Optimal Scaling of Random Walk Metropolis Algorithms," *The Annals of Applied Probability*, vol. 7, pp. 110–120, 1997.

[151] J. M. Flegal and G. L. Jones, "Batch means and spectral variance estimators in Markov chain Monte Carlo," *The Annals of Statistics*, vol. 38, pp. 1034–1070, 2010.

[152] C. Geyer, "Introduction to Markov chain Monte Carlo," *Chapman and Hall/CRC Handbooks of Modern Statistical Methods*, vol. 1, 2011.

[153] M. Bedard, "Optimal Acceptance Rates for Metropolis Algorithms: Moving beyond 0.234," *Stochastic Processes and Their Applications*, vol. 118, p. 2198–2222, 2008.

[154] G. Casella and E. I. George, "Explaining the gibbs sampler," *The American Statistician*, vol. 46, pp. 167–174, 1992.

[155] H. Haario, E. Saksman, and J. Tamminen, "An adaptive Metropolis algorithm," *Bernoulli*, vol. 7, pp. 223–242, 2001.

[156] S. Chib and E. Greenberg, "Understanding the Metropolis-Hastings Algorithm," *Journal of Engineering Mechanics*, vol. 49, p. 327, 1995.

[157] R. Costilla, I. Liu, R. Arnold, and D. Fernandez, "Bayesian Model-based clustering for Longitudinal ordinal data," *Computational Statistics*, vol. 34, p. 1015–1038, 2019.

[158] W. R. Gilks, N. G. Best, and K. K. Tan, "Adaptive Rejection Metropolis Sampling within Gibbs Samplings," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 44, pp. 455–472, 1995.

[159] A. A. Johnson, G. L. Jones, and R. C. Neath, "Component-Wise Markov Chain Monte Carlo: Uniform and Geometric Ergodicity under Mixing and Composition," *Statistical Science*, vol. 28, pp. 360–375, 2013.

[160] H. Haario, E. Saksman, and J. Tamminen, "Componentwise adaptation for high dimensional MCMC," *Computational Statistics*, vol. 20, p. 265–273, 2005.

[161] R. A. Levine, Z. Yu, W. G. Hanley, and J. J. Nitao, "Implementing componentwise Hastings algorithms," *Computational Statistics and Data Analysis*, vol. 48, pp. 363–389, 2005.

[162] P. Honarmandi, L. Johnson, and R. Arroyave, "Bayesian probabilistic prediction of precipitation behavior in Ni-Ti shape memory alloys," *Computational Materials Science*, vol. 172, p. 109334, 2020.

[163] K. Kawahara, R. Ishikawa, T. Higashi, T. Kimura, Y. H. Ikuhara, N. Shibata, and Y. Ikuhara, "Unique fitting of electrochemical impedance spectra by random walk Metropolis Hastings algorithm," *Journal of Power Sources*, vol. 403, p. 184–191, 2018.

[164] J. S. Teixeira, L. T. Stutz, D. C. Knupp, and A. J. Silva-Neto, "A new adaptive approach of the Metropolis-Hastings algorithm applied to structural damage identification using time domain data," *Applied Mathematical Modelling*, vol. 82, p. 587–606, 2020.

[165] P. Grišins and I. E. Mazets, "Metropolis–Hastings thermal state sampling for numerical simulations of Bose–Einstein condensates," *Computer Physics Communications*, vol. 185, p. 1926–1931, 2014.

[166] S. Biswal and A. Ramaswamy, "Finite element model updating of concrete structures based on imprecise probability," *Mechanical Systems and Signal Processing*, vol. 94, pp. 165–179, 2017.

[167] P. Green, E. Cross, and K. Worden, "Bayesian system identification of dynamical systems using highly informative training data," *Mechanical Systems and Signal Processing*, vol. 56-57, pp. 109–122, 2015.

[168] J. L. Beck and S. K. Au, "Bayesian Updating of Structural Models and Reliability using Markov Chain Monte Carlo Simulation," *Journal of Engineering Mechanics*, vol. 128, pp. 380–391, 2002.

[169] W. Betz, I. Papaioannou, and D. Straub, “Transitional Markov Chain Monte Carlo: Observations and Improvements,” *Journal of Engineering Mechanics*, vol. 142, p. , 2016.

[170] S. K. Au and J. L. Beck, “Estimation of small failure probabilities in high dimensions by subset simulation,” *Probabilistic Engineering Mechanics*, vol. 16, pp. 263–277, 2001.

[171] J. Y. Ching and J. S. Wang, “Application of the transitional Markov chain Monte Carlo algorithm to probabilistic site characterization,” *Engineering Geology*, vol. 203, pp. 151–167, 2016.

[172] G. A. Ortiz, D. A. Alvarez, and D. Bedoya-Ruíz, “Identification of Bouc–wen type models using the Transitional Markov Chain Monte Carlo method,” *Computers and Structures*, vol. 146, pp. 252–269, 2015.

[173] W. H. Zhou, F. Tan, and K. V. Yuen, “Model updating and uncertainty analysis for creep behavior of soft soil,” *Computers and Geotechnics*, vol. 100, pp. 135–143, 2018.

[174] J. Wang and L. Katafygiotis, “Reliability-based optimal design of linear structures subjected to stochastic excitations,” *Structural Safety*, vol. 47, pp. 29–38, 2014.

[175] H. M. Gomes, M. Broggi, E. Patelli, and J. E. Mottershead, “Model Updating by Uncertain Parameter Inference,” *Vulnerability, Uncertainty, and Risk*, vol. , p. , 2014.

[176] O. Sedehi, C. Papadimitriou, and L. S. Katafygiotis, “Probabilistic hierarchical bayesian framework for time-domain model updating and robust predictions,” *Mechanical Systems and Signal Processing*, vol. 123, pp. 648–673, 2019.

[177] A. Doucet, N. de Freitas, N. Gordon, and A. Smith, *Sequential Monte Carlo methods in practice.* New York: Springer, 2001. ISBN: 978-0387951461.

[178] O. Cappe, S. J. Godsill, and E. Moulines, “An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo,” *Proceedings of the IEEE*, vol. 95, pp. 899–924, 2007.

[179] T. J. Rogers, K. Worden, and E. J. Cross, “Bayesian Solutions to State-Space Structural Identification,” *Model Validation and Uncertainty Quantification*, vol. 3, pp. 247–253, 2020.

[180] T. J. Rogers, T. B. Schon, A. Lindholm, K. Worden, and E. J. Cross, “Identification of a Duffing oscillator using particle Gibbs with ancestor sampling,” *Journal of Physics: Conference Series*, vol. 1264, p. 012051, 2019.

[181] N. Chopin, “A Sequential Particle Filter Method for Static Models,” *Biometrika*, vol. 89, pp. 539–552, 2002.

[182] M. Beer, I. A. Kougioumtzoglou, and E. Patelli, “Emerging Concepts and Approaches for Efficient and Realistic Uncertainty Quantification,” *Maintenance and Safety of Aging Infrastructure*, vol. , pp. 121–162, 2014.

[183] R. M. Neal, “Annealed importance sampling,” *Statistics and Computing*, vol. 11, p. 125–139, 2001.

[184] G. Zhu, X. Li, J. Ma, Y. Wang, S. Liu, C. Huang, K. Zhang, and X. Hu, “A new moving strategy for the Sequential Monte Carlo approach in optimizing the hydrological model parameters,” *Advances in Water Resources*, vol. 114, p. 164–179, 2018.

[185] J. M. Hammersley and D. C. Handscomb, *Monte Carlo methods*. Springer Netherlands, 1964. ISBN: 978-9400958197.

[186] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo Method*. John Wiley & Sons, 3 ed., 2016. ISBN: 978-1118632161.

[187] M. Sambridge, “Geophysical inversion with a neighbourhood algorithm — I. Searching a parameter space,” *Geophysical Journal International*, vol. 138, p. 479–494, 1999.

[188] L. M. Murray, A. Lee, and P. E. Jacob, “Parallel resampling in the particle filter,” *Journal of Computational and Graphical Statistics*, vol. 25, p. 789–805, 2016.

[189] T. B. Schön, A. Svensson, L. Murray, and F. Lindsten, “Probabilistic learning of nonlinear dynamical systems using Sequential Monte Carlo,” *Mechanical Systems and Signal Processing*, vol. 104, pp. 866–883, 2018.

[190] J. Sun, H. Zuo, W. Wang, and M. G. Pecht, “Prognostics uncertainty reduction by fusing on-line monitoring data based on a state-space-based degradation model,” *Mechanical Systems and Signal Processing*, vol. 45, pp. 396–407, 2014.

[191] A. Svensson, T. B. Schon, and F. Lindsten, "Learning of state-space models with highly informative observations: A tempered sequential Monte Carlo solution," *Mechanical Systems and Signal Processing*, vol. 104, pp. 915–928, 2018.

[192] D. Wang, S. Sun, and P. W. Tse, "A general sequential Monte Carlo method based optimal wavelet filter: A Bayesian approach for extracting bearing fault features," *Mechanical Systems and Signal Processing*, vol. 52-53, pp. 293–308, 2015.

[193] M. Chen, R. Lin, T. Y. Ng, and F. Ding, "Particle filter-based algorithm of simultaneous output and parameter estimation for output nonlinear systems under low measurement rate constraints," *Nonlinear Dynamics*, vol. 107, p. 727–741, 2021.

[194] J. H. Yang and H. F. Lam, "An efficient adaptive Sequential Monte Carlo method for Bayesian model updating and damage detection," *Structural Control and Health Monitoring*, vol. 25, 2018.

[195] C. Sbarufatti, M. Corbetta, A. Manes, and M. Giglio, "Sequential Monte-Carlo sampling based on a committee of artificial neural networks for posterior state estimation and residual lifetime prediction," *International Journal of Fatigue*, vol. 83, p. 10–23, 2016.

[196] M. Corbetta, G. Sbarufatti, A. Manes, and M. Giglio, "Real-time prognosis of crack growth evolution using Sequential Monte Carlo Methods and statistical model parameters," *IEEE Transactions on Reliability*, vol. 64, p. 736–753, 2015.

[197] A. Y. Pisal and R. S. Jangid, "Dynamic Response of Structure with Tuned Mass Friction Damper," *International Journal of Advanced Structural Engineering*, vol. 8, p. 363–377, 2016.

[198] J. L. Beck and S. K. Au, "Bayesian Updating of Structural Models and Reliability using Markov Chain Monte Carlo Simulation," *Journal of Engineering Mechanics*, vol. 128, pp. 380–391, 2002.

[199] E. Balmes, "Garteur Group on Ground Vibration Testing: Results from the test of a single structure by 12 laboratories in Europe," *In Proceedings of the International Modal Analysis Conference*, pp. 1346–1352, 1997.

[200] M. Broggi, M. Faes, E. Patelli, Y. Govers, D. Moens, and M. Beer, "Comparison of Bayesian and interval uncertainty quantification: Application to the AIRMOD test

structure," *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, vol. , p. , 2017.

[201] J. E. Mottershead, M. Link, and M. I. Friswell, "The Sensitivity Method in Finite Element Model Updating: A Tutorial," *Mechanical Systems and Signal Processing*, vol. 25, p. 2275–2296, 2011.

[202] J. E. Mottershead and M. I. Friswell, "Model Updating In Structural Dynamics: A Survey," *Journal of Sound and Vibration*, vol. 167, p. 347–375, 1993.

[203] E. Patelli, "COSSAN: A Multidisciplinary Software Suite for Uncertainty Quantification and Risk Management," *Handbook of Uncertainty Quantification*, vol. , pp. 1909–1977, 2017.

[204] E. Patelli, M. Broggi, M. D. Angelis, and M. Beer, "OpenCossan: An Efficient Open Tool for Dealing with Epistemic and Aleatory Uncertainties," *Vulnerability, Uncertainty, and Risk*, vol. , p. , 2014.

[205] F. Massey, "The Kolmogorov-Smirnov Test for Goodness of Fit," *Journal of the American Statistical Association*, vol. 46, pp. 68–78, 1951.

[206] G. Marsaglia, W. W. Tsang, and J. Wang, "Evaluating Kolmogorovs Distribution," *Journal of Statistical Software*, vol. 8, 2003.

[207] M. A. Stephens, "Introduction to Kolmogorov (1933) On the Empirical Determination of a Distribution," *Springer Series in Statistics Breakthroughs in Statistics*, vol. , pp. 93–105, 1992.

[208] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson, "Obstacles to High-Dimensional Particle Filtering," *Monthly Weather Review*, vol. 136, pp. 4629–4640, 2008.

[209] P. Rebeschini and R. V. Handel, "Can local particle filters beat the curse of dimensionality?," *The Annals of Applied Probability*, vol. 25, pp. 2809–2866, 2015.

[210] S. K. Au and J. L. Beck, "Important Sampling in High Dimensions," *Structural Safety*, vol. 25, p. 139–163, 2003.

[211] G. O. Roberts and J. S. Rosenthal, "Coupling and Ergodicity of Adaptive Markov Chain Monte Carlo Algorithms," *Journal of Applied Probability*, vol. 44, pp. 458–475, 2007.

[212] G. O. Roberts and J. S. Rosenthal, “Examples of Adaptive MCMC,” *Journal of Computational and Graphical Statistics*, vol. 18, pp. 349–367, 2009.

[213] H. F. Lam, J. Yang, and S. K. Au, “Bayesian model updating of a coupled-slab system using field test data utilizing an enhanced Markov Chain Monte Carlo simulation algorithm,” *Engineering Structures*, vol. 102, p. 144–155, 2015.

[214] N. Kantas, A. Beskos, and A. Jasra, “Sequential Monte Carlo Methods for High-Dimensional Inverse Problems: A Case Study for the Navier-Stokes Equations,” *SIAM/ASA Journal on Uncertainty Quantification*, vol. 2, pp. 464–489, 2014.

[215] C. A. Naesseth, F. Lindsten, and T. B. Schön, “Nested Sequential Monte Carlo Methods,” *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, vol. 37, p. 1292–1301, 2015.

[216] C. A. Naesseth, F. Lindsten, and T. B. Schön, “High-Dimensional Filtering Using Nested Sequential Monte Carlo,” *IEEE Transactions on Signal Processing*, vol. 67, pp. 4177–4188, 2019.

[217] P. L. Green, L. J. Devlin, R. E. Moore, R. J. Jackson, J. Li, and S. Maskell, “Increasing the efficiency of Sequential monte carlo samplers through the use of approximately optimal L-kernels,” *Mechanical Systems and Signal Processing*, vol. 162, p. 108028, 2022.

[218] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A Tutorial on Particle Filters for On-line Nonlinear/Non-gaussian Bayesian Tracking,” *IEEE Transactions on Signal Processing*, vol. 50, pp. 174–188, 2002.

[219] H. D. Vo, Z. Fox, A. Baetica, and B. Munsky, “Bayesian estimation for stochastic gene expression using multifidelity models,” *The Journal of Physical Chemistry B*, vol. 123, pp. 2217–2234, 2019.

[220] M. N. Chatzis and E. N. Chatzi, “Online Bayesian Identification of Non-Smooth Systems,” *Procedia Engineering*, vol. 199, pp. 918–923, 2017.

[221] G. Ou, S. J. Dyke, and A. Prakash, “Real time hybrid simulation with online model updating: An analysis of accuracy,” *Mechanical Systems and Signal Processing*, vol. 84B, pp. 223–240, 2017.

[222] J. H. Weng and C. H. Loh, “Recursive subspace identification for on-line tracking of structural modal parameter,” *Mechanical Systems and Signal Processing*, vol. 25, pp. 2923–2937, 2011.

[223] J. Chen, S. Yuan, and H. Wang, “On-line updating gaussian process measurement model for crack prognosis using the particle filter,” *Mechanical Systems and Signal Processing*, vol. 140, p. 106646, 2020.

[224] S. Mallapur and R. Platz, “Uncertainty quantification in the mathematical modelling of a suspension strut using Bayesian inference,” *Mechanical Systems and Signal Processing*, vol. 118, pp. 158–170, 2019.

[225] R. Fuentes, N. Dervilis, K. Worden, and E. Cross, “Efficient parameter identification and model selection in nonlinear dynamical systems via sparse Bayesian learning,” *Journal of Physics: Conference Series*, vol. 1264, p. 012050, 2019.

[226] S. Sehgal and H. Kumar, “Structural Dynamic Model Updating Techniques: A State of the Art Review,” *Archives of Computational Methods in Engineering*, vol. 23, pp. 515–533, 2015.

[227] A. Gray, A. Wimbush, M. D. Angelis, P. O. Hristov, E. Miralles-Dolz, D. Calleja, and R. Rocchetta, “From inference to design: A comprehensive framework for uncertainty quantification in engineering with limited information,” *Mechanical Systems and Signal Processing*, vol. 165, p. 108210, 2022.

[228] K. Yuen, “Recent developments of Bayesian model class selection and applications in civil engineering,” *Structural Safety*, vol. 32, pp. 338–346, 2010.

[229] H. Jensen and D. Jerez, “A stochastic framework for hydraulic performance assessment of complex water distribution networks: Application to connectivity detection problems,” *Probabilistic Engineering Mechanics*, vol. 60, p. , 2020.

[230] S. E. Minson, M. Simons, and J. L. Beck, “Bayesian inversion for Finite Fault earthquake source models I — theory and algorithm,” *Geophysical Journal International*, vol. 194, p. 1701–1726, 2013.

[231] D. Foreman-Mackey, D. W. Hogg, D. Lang, and J. Goodman, “Emcee: The mcmc hammer,” *Publications of the Astronomical Society of the Pacific*, vol. 125, pp. 306–312, 2013.

[232] C. Andrieu and J. Thoms, “A tutorial on adaptive MCMC,” *Statistics and Computing*, vol. 18, pp. 343–373, 2008.

[233] L. Martino and V. Elvira, “Metropolis sampling,” *Wiley StatsRef: Statistics Reference Online*, vol. , pp. 1–18, 2017.

[234] S. M. Lynch, *Introduction to applied Bayesian statistics and estimation for social scientists*. Springer Science and Business Media, 2007. ISBN: 978-0387712659.

[235] Y. Zhang and W. Yang, “A comparative study of the stochastic simulation methods applied in structural health monitoring,” *Engineering Computations*, vol. 31, pp. 1484–1513, 2014.

[236] A. Lye, A. Cicirello, and E. Patelli, “Bayesian Model Updating of Reliability Parameters using Transitional Markov Chain Monte Carlo with Slice Sampling,” *In Proceedings of the 30th European Safety and Reliability Conference and 15th Probabilistic Safety Assessment and Management Conference*, vol. , p. , 2020.

[237] P. Angelikopoulos, C. Papadimitriou, and P. Koumoutsakos, “X-TMCMC: Adaptive kriging for Bayesian inverse modeling,” *Computer Methods in Applied Mechanics and Engineering*, vol. 289, pp. 409–428, 2015.

[238] Y. Jin, Z. Yin, W. Zhou, and S. Horpibulsuk, “Identifying parameters of Advanced Soil models using an enhanced transitional Markov chain Monte Carlo method,” *Acta Geotechnica*, vol. 14, pp. 1925–1947, 2019.

[239] F. Hou, J. Goodman, D. W. Hogg, J. Weare, and C. Schwab, “An Affine-Invariant Sampler For Exoplanet Fitting And Discovery In Radial Velocity Data,” *The Astrophysical Journal*, vol. 745, p. 198, 2012.

[240] I. J. M. Crossfield, E. Petigura, J. E. Schlieder, A. W. Howard, B. J. Fulton, K. M. Aller, D. R. Ciardi, S. Lépine, T. Barclay, I. D. Pater, K. D. Kleer, E. V. Quintana, J. L. Christiansen, E. Schlafly, L. Kaltenegger, J. R. Crepp, T. Henning, C. Obermeier, N. Deacon, L. Weiss, H. T. Isaacson, B. M. S. Hansen, M. C. Liu, T. Greene, S. B. Howell, T. Barman, and C. Mordasini, “A Nearby M Star With Three Transiting Super-Earths Discovered By K2,” *The Astrophysical Journal*, vol. 804, p. 10, 2015.

[241] A. Vanderburg, B. T. Montet, J. A. Johnson, L. A. Buchhave, L. Zeng, F. Pepe, A. C. Cameron, D. W. Latham, E. Molinari, S. Udry, C. Lovis, J. M. Matthews, C. Cameron,

N. Law, B. P. Bowler, R. Angus, C. Baranec, A. Bieryla, W. Boschin, D. Charbonneau, R. Cosentino, X. Dumusque, P. Figueira, D. B. Guenther, A. Harutyunyan, C. Hellier, R. Kuschnig, M. Lopez-Morales, M. Mayor, G. Micela, A. F. J. Moffat, M. Pedani, D. F. Phillips, G. Piotto, D. Pollacco, D. Queloz, K. Rice, R. Riddle, J. F. Rowe, S. M. Rucinski, D. Sasselov, D. Ségransan, A. Sozzetti, A. Szentgyorgyi, C. Watson, and W. W. Weiss, "Characterizing K2 Planet Discoveries: A Super-Earth Transiting The Bright K Dwarf Hip 116454," *The Astrophysical Journal*, vol. 800, p. 59, 2015.

[242] S. Luszcz-Cook, K. D. Kleer, I. D. Pater, M. Adamkovics, and H. Hammel, "Retrieving Neptune's aerosol properties from Keck OSIRIS observations. I. Dark regions," *Icarus*, vol. 276, pp. 52–87, 2016.

[243] Y. Zhou, A. D. Popolo, and Z. Chang, "On the absence of a universal surface density, and a maximum Newtonian acceleration in dark matter haloes: Consequences for MOND," *Physics of the Dark Universe*, vol. 28, p. , 2020.

[244] Y. Matviychuk, E. V. Harbou, and J. D. Holland, "An experimental validation of a Bayesian model for quantification in NMR spectroscopy," *Journal of Magnetic Resonance*, vol. 285, pp. 86–100, 2017.

[245] S. Lunderman, M. Morzfeld, F. Glassmeier, and G. Feingold, "Estimating parameters of the nonlinear cloud and rain equation from a large-eddy simulation," *Physica D: Nonlinear Phenomena*, vol. 410, p. , 2020.

[246] L. Alawieh, J. Goodman, and J. B. Bell, "Iterative construction of Gaussian process surrogate models for Bayesian inference," *Journal of Statistical Planning and Inference*, vol. 207, pp. 55–72, 2020.

[247] J. Zhang and M. D. Shields, "On the quantification and efficient propagation of imprecise probabilities resulting from small datasets," *Mechanical Systems and Signal Processing*, vol. 98, pp. 465–483, 2018.

[248] N. H. Paulson, B. J. Bocklund, R. A. Otis, Z. K. Liu, and M. Stan, "A Stochastic Regression Approach to Analyzing Thermodynamic Uncertainty in Chemical Speciation Modeling," *Acta Materialia*, vol. 174, pp. 9–15, 2019.

[249] F. P. Hollick, V. Gori, and C. A. Elwell, "Thermal performance of occupied homes: A

dynamic grey-box method accounting for solar gains," *Energy and Buildings*, vol. 208, p. , 2020.

[250] P. Wagner, R. Fahrni, M. Klippel, A. Frangi, and B. Sudret, "Bayesian calibration and sensitivity analysis of heat transfer models for fire insulation panels," *Engineering Structures*, vol. 205, p. , 2020.

[251] J. Zhang, S. Termaath, and M. D. Shields, "Imprecise global sensitivity analysis using bayesian multimodel inference and importance sampling," *Mechanical Systems and Signal Processing*, vol. 148, p. , 2021.

[252] J. H. Gallier, *Geometric methods and applications: For computer science and engineering*. Springer Science and Business Media, 2012. ISBN: 978-1461301370.

[253] Y. Wang and J. Solomon, "Intrinsic and extrinsic operators for shape analysis," *Handbook of Numerical Analysis Processing, Analyzing and Learning of Images, Shapes, and Forms: Part 2*, vol. , pp. 41–115, 2019.

[254] T. Lampart, "Implementation and performance comparison of an ensemble sampler with affine invariance," technical Report, MOSAIC Group, Institute of Theoretical Computer Science, Department of Computer Science, ETH Zurich, 2012.

[255] P. H. Garthwaite, Y. Fan, and S. A. Sisson, "Adaptive optimal scaling of Metropolis–Hastings algorithms using the Robbins–Monro process," *Communications in Statistics - Theory and Methods*, vol. 45, pp. 5098–5111, 2016.

[256] G. O. Roberts and J. S. Rosenthal, "Examples of Adaptive MCMC," *Journal of Computational and Graphical Statistics*, vol. 18, pp. 349–367, 2009.

[257] C. Safta, M. Khalil, and H. N. Najm, "Transitional Markov Chain Monte Carlo Sampler in UQTk," *SANDIA Report*, vol. , p. , 2020.

[258] A. F. Agarap, "Deep Learning using Rectified Linear Units (ReLU)," *ArXiv*, vol. abs/1803.08375, 2018.

[259] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[260] J. J. Jenq and W. Li, "Feedforward backpropagation artificial neural networks on reconfigurable meshes," *Future Generation Computer Systems*, vol. 14, pp. 313–319, 1998.

[261] F. Ruggeri, D. R. Insua, and J. Martín, "Robust bayesian analysis," *Handbook of Statistics Bayesian Thinking - Modeling and Computation*, vol. , pp. 623–667, 2005.

[262] J. O. Berger, E. Moreno, and L. R. P. *et al.*, "An overview of robust Bayesian analysis," *Test*, vol. 3, p. 5–124, 1994.

[263] L. G. Crespo, S. P. Kenny, and D. P. Giesy, "The NASA Langley Multidisciplinary Uncertainty Quantification Challenge," *In the Proceedings of the 16th AIAA Non-Deterministic Approaches Conference*, vol. 1, 2014.

[264] A. Lye, M. Kitahara, M. Broggi, and E. Patelli, "Robust optimization of a dynamic Black-box system under severe uncertainty: A distribution-free framework," *Mechanical Systems and Signal Processing*, vol. 167, p. 108522, 2022.

[265] E. Patelli, D. A. Alvarez, M. Broggi, and M. D. Angelis, "Uncertainty management in multidisciplinary design of critical safety systems," *Journal of Aerospace Information Systems*, vol. 12, pp. 140–169, 2015.

[266] B. M. Turner and T. V. Zandt, "A tutorial on approximate Bayesian computation," *Journal of Mathematical Psychology*, vol. 56, no. 2, pp. 69–85, 2012.

[267] S. Bi, M. Broggi, and M. Beer, "The role of the Bhattacharyya distance in stochastic model updating," *Mechanical Systems and Signal Processing*, vol. 117, pp. 437–452, 2019.

[268] M. Heideman, D. Johnson, and C. Burrus, "Gauss and the history of the fast fourier transform," *IEEE ASSP Magazine*, vol. 1, pp. 14–21, 1984.

[269] Y. Bai, Z. Huang, and H. Lam, "A Distributionally Robust Optimization Approach to the Nasa Langley Uncertainty Quantification Challenge," *In Proceedings of the* 30^{th} *European Safety and Reliability Conference and the* 15^{th} *Probabilistic Safety Assessment and Management Conference*, vol. 1, 2020.

[270] V. M. Panaretos and Y. Zemel, "Statistical aspects of wasserstein distances," *Annual Review of Statistics and Its Application*, vol. 6, pp. 405–431, 2019.

[271] S. Ferson, W. Oberkampf, and L. Ginzburg, "Model validation and predictive capability for the thermal challenge problem," *Computer Methods in Applied Mechanics and Engineering*, vol. 197, pp. 2408–2430, 2008.

[272] L. G. Crespo, S. P. Kenny, D. P. Giesy, and B. K. Stanford, "Random variables with moment-matching staircase density functions," *Applied Mathematical Modelling*, vol. 64, pp. 196–213, 2018.

[273] M. Kitahara, S. F. Bi, M. Broggi, and M. Beer, "Bayesian Model Updating in Time Domain with Metamodel-based Reliability Method," *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, vol. 7, 2021.

[274] A. W. Bowman and A. Azzalini, *Applied Smoothing Techniques for Data Analysis.* OUP Oxford, 1997. ISBN: 978-0191545696.

[275] D. Dubois, L. Foulloy, G. Mauris, and H. M. Prade, "Probability-possibility transformations, triangular fuzzy sets, and probabilistic inequalities," *Reliable Computing*, vol. 10, p. 273–297, 2004.

[276] D. Saad, *On-Line Learning in Neural Networks.* Cambridge: Cambridge University Press, 2009. ISBN: 978-0521117913.

[277] D. Hu, W. Wang, X. Zhang, and K. Chen, "On-line real-time mistuning identification and model calibration method for rotating blisks based on blade tip timing (BTT)," *Mechanical Systems and Signal Processing*, vol. 147, p. 107074, 2021.

[278] N. Zhang, T. Chen, M. Zheng, L. Luo, and P. Liu, "Real-time identification of vehicle body motion-modes based on motion-mode energy method," *Mechanical Systems and Signal Processing*, vol. 143, p. 106843, 2020.

[279] Z. Xie and J. Feng, "Real-time nonlinear structural system identification via iterated unscented Kalman filter," *Mechanical Systems and Signal Processing*, vol. 28, pp. 309–322, 2012.

[280] Y. Li, L. Ding, Z. Zheng, Q. Yang, X. Zhao, and G. Liu, "A multi-mode real-time terrain parameter estimation method for wheeled motion control of mobile robots," *Mechanical Systems and Signal Processing*, vol. 104, pp. 758–775, 2018.

[281] P. Ghaderi and F. Amini, "Development of a new method for online parameter identification in seismically excited smart building structures using virtual synchronization and adaptive control design," *Applied Mathematical Modelling*, vol. 87, pp. 203–221, 2020.

[282] S. Sarrka, *Bayesian filtering and smoothing.* Cambridge: Cambridge University Press, 4 ed., 2013. ISBN: 978-1107030657.

[283] X. Fan and Y. Liu, "Use of monitored daily extreme stress data for performance prediction of steel bridges: Dynamic linear models and Gaussian mixed particle filter," *Mechanical Systems and Signal Processing*, vol. 121, pp. 841–855, 2019.

[284] L. Morse, Z. S. Khodaei, and M. H. Aliabadi, "Reliability based impact localization in composite panels using Bayesian updating and the Kalman filter," *Mechanical Systems and Signal Processing*, vol. 99, pp. 107–128, 2018.

[285] J. Huang, X. Li, F. Zhang, and Y. Lei, "Identification of joint structural state and earthquake input based on a generalized Kalman filter with unknown input," *Mechanical Systems and Signal Processing*, vol. 151, p. 107362, 2021.

[286] A. Lund, S. J. Dyke, W. Song, and I. Bilionis, "Identification of an experimental nonlinear energy sink device using the unscented Kalman filter," *Mechanical Systems and Signal Processing*, vol. 136, p. 106512, 2020.

[287] R. Astroza, A. Alessandri, and J. P. Conte, "A dual adaptive filtering approach for nonlinear finite element model updating accounting for modeling uncertainty," *Mechanical Systems and Signal Processing*, vol. 115, pp. 782–800, 2019.

[288] X. Wang, X. He, J. R. Williams, R. C. Izaurralde, and J. D. Atwood, "Sensitivity And Uncertainty Analyses Of Crop Yields And Soil Organic Carbon Simulated With Epic," *Transactions of the ASAE*, vol. 48, no. 3, pp. 1041–1054, 2005.

[289] G. Terejanu, P. Singla, T. Singh, and P. D. Scott, "Adaptive Gaussian Sum Filter for Nonlinear Bayesian Estimation," *IEEE Transactions on Automatic Control*, vol. 56, pp. 2151–2156, 2011.

[290] J. Chen, S. Yuan, and X. Jin, "On-line prognosis of fatigue cracking via a regularized particle filter and guided wave monitoring," *Mechanical Systems and Signal Processing*, vol. 131, pp. 1–17, 2019.

[291] H. Rozas, F. Jaramillo, A. Perez, D. Jimenez, M. E. Orchard, and K. Medjaher, "A method for the reduction of the computational cost associated with the implementation of particle-filter-based failure prognostic algorithms," *Mechanical Systems and Signal Processing*, vol. 135, p. 106421, 2020.

[292] E. A. Wan and R. V. D. Merwe, *Kalman Filtering and Neural Network*. Wiley, New York, 2002.

[293] H. Kim, C. Jin, and M. Kim, "Real-time estimation of riser's deformed shape using inclinometers and Extended Kalman Filter," *Marine Structures*, vol. 77, p. 102933, 2021.

[294] S. Hur, "Short-term wind speed prediction using Extended Kalman filter and machine learning," *Energy Reports*, vol. 7, pp. 1046–1054, 2021.

[295] G. Terejanu, P. Singla, T. Singh, and P. D. Scott, "A novel gaussian sum filter method for accurate solution to the nonlinear filtering problem," *In Proceedings of the 11th International Conference on Information Fusion*, vol. 1, pp. 1–8, 2008.

[296] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Boston: Artech House, 2004. ISBN: 978-1580538510.

[297] K. E. Tatis, V. K. Dertimanis, and E. N. Chatzi, "Sequential Bayesian inference for Uncertain Nonlinear Dynamic Systems: A tutorial," *Journal of Structural Dynamics*, vol. 1, pp. 236–262, 2022.

[298] J. Olsson, O. Cappe, R. Douc, and E. Moulines, "Sequential monte carlo smoothing with application to parameter estimation in nonlinear state space models," *Bernoulli*, vol. 14, pp. 155–179, 2008.

[299] O. Frank, J. Nieto, J. Guivant, and S. Scheding, "Multiple target tracking using Sequential Monte Carlo Methods and statistical data association," *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, vol. 3, pp. 2718–2723, 2003.

[300] A. Lye, L. Marino, A. Cicirello, and E. Patelli, "Sequential Ensemble Monte Carlo Sampler for On-line Bayesian Inference of Time-varying Parameter in Engineering Applications," *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems Part B: Mechanical Engineering*, pp. 1–13, 2023.

[301] M. D. Shields, D. G. Giovanis, and V. S. Sundar, “Subset simulation for problems with strongly non-Gaussian, highly anisotropic, and degenerate distributions,” *Computers and Structures*, vol. 245, p. 106431, 2021.

[302] T. B. Schon, F. Lindsten, J. Dahlin, J. Wagberg, C. A. Naesseth, A. Svensson, and L. Dai, “Sequential Monte Carlo Methods for System Identification,” *IFAC-PapersOnLine*, vol. 48, pp. 775–786, 2015.

[303] S. Eftekhar-Azam, S. Mariani, and N. K. A. Attari, “Online damage detection via a synergy of proper orthogonal decomposition and recursive bayesian filters,” *Nonlinear Dynamics*, vol. 89, p. 1489–1511, 2017.

[304] B. W. Silverman, D. R. Cox, and D. V. Hinkley, *Density Estimation for Statistics and Data Analysis.* Taylor and Francis, 1 ed., 1986.

[305] C. C. Drovandi, J. M. McGree, and A. N. Pettitt, “A Sequential Monte Carlo Algorithm to Incorporate Model Uncertainty in Bayesian Sequential Design,” *Journal of Computational and Graphical Statistics*, vol. 23, pp. 3–24, 2014.

[306] T. Toni, D. Welch, N. Strelkowa, A. Ipsen, and M. P. Stumpf, “Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems,” *Journal of the Royal Society Interface*, vol. 6, p. 187–202, 2008.

[307] T. L. T. Nguyen, F. Septier, G. W. Peters, and Y. Delignon, “Bayesian model selection and parameter estimation in penalized regression model using SMC samplers,” *In Proceedings of 21st European Signal Processing Conference (EUSIPCO 2013)*, vol. 1, pp. 1–5, 2013.

[308] I. Urteaga, M. F. Bugallo, and P. M. Djuric, “Sequential Monte Carlo methods under model uncertainty,” *In Proceedings of 2016 IEEE Statistical Signal Processing Workshop (SSP)*, vol. , pp. 1–5, 2016.

[309] Y. Zhou, A. M. Johansen, and J. A. D. Aston, “Toward Automatic Model Comparison: An Adaptive Sequential Monte Carlo Approach,” *Journal of Computational and Graphical Statistics*, vol. 25, pp. 701–726, 2016.

[310] P. D. Moral, A. Doucet, and A. Jasra, “Sequential Monte Carlo for Bayesian Computations,” *Bayesian Statistics*, vol. 8, pp. 1–34, 2007.

[311] G. Moffa and J. Kuipers, "Sequential Monte Carlo EM for multivariate probit models," *Computational Statistics and Data Analysis*, vol. 72, pp. 252–272, 2014.

[312] P. D. Moral, A. Doucet, and A. Jasra, "On adaptive resampling strategies for sequential Monte Carlo methods," *Bernoulli*, vol. 18, pp. 252–278, 2012.

[313] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, p. 341–359, 1997.

[314] M. Bognanni and J. Zito, "Sequential Bayesian inference for vector autoregressions with stochastic volatility," *Journal of Economic Dynamics and Control*, vol. 113, p. 103851, 2020.

[315] L. Marino, A. Cicirello, and D. A. Hills, "Displacement transmissibility of a Coulomb friction oscillator subject to joined base-wall motion," *Nonlinear Dynamics*, vol. 98, p. 2595–2612, 2019.

[316] J. P. Den-Hartog, "Forced vibrations with combined viscous and Coulomb damping," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 9, pp. 801–817, 1930.

[317] C. V. Loan, *Computational Frameworks for the Fast Fourier Transform.* SIAM, 1992. ISBN: 978-0898712858.

[318] A. Kong, J. S. Liu, and W. H. Wong, "Sequential Imputations and Bayesian Missing Data Problems," *Journal of the American Statistical Association*, vol. 89, pp. 278–288, 1994.

[319] K. H. Li, "The Sampling/Importance Resampling Algorithm," *Applied Bayesian Modeling and Causal Inference from Incomplete-Data Perspectives*, vol. , pp. 265–276, 2004.

[320] N. Prinja, "Artificial Intelligence in Nuclear," *American Society for Quality Winter 2022 Energy and Environmental Division Newsletter*, vol. 1, p. 18–22, 2022.

[321] N. Prinja, "Artificial Intelligence for Engineering Projects," Oct 2021.

[322] J. J. Berman, "Understanding your data," *Data Simplification*, pp. 135–187, 2016.

[323] H. Hu, “vReLU Activation Functions for Artificial Neural Networks,” *In the Proceedings of the 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, vol. 1, pp. 856–860, 2018.

[324] H. Smith and N. R. Draper, *Applied Regression Analysis.* Wiley, 1998.

[325] S. Tolo, X. Tian, N. Bausch, V. Becerra, T. V. Santhosh, G. Vinod, and E. Patelli, “Robust on-line diagnosis tool for the early accident detection in nuclear power plants,” *Reliability Engineering and System Safety*, vol. 186, pp. 110–119, 2019.

[326] X. Wang and Y. Zhou, “Solution of the Static Deflection Mode Shape Function of the Cantilever Beam under Transverse Flow Based on the Boundary Shooting Method,” *Science and Technology of Nuclear Installations*, vol. 2021, pp. 1–8, 2021.

[327] M. S. Eldred, H. Agarwal, V. M. Perez, S. F. Wojtkiewicz-Jr, and J. E. Renaud, “Investigation of reliability method formulations in DAKOTA/UQ,” *Structure and Infrastructure Engineering*, vol. 3, pp. 199–213, 2007.

[328] J. Nakajima, M. Kasuya, and T. Watanabe, “Bayesian analysis of time-varying parameter vector autoregressive model for the Japanese economy and monetary policy,” *Journal of the Japanese and International Economies*, vol. 25, pp. 225–245, 2011.

[329] S. Dimitrakopoulos, “Semiparametric bayesian inference for time-varying parameter regression models with stochastic volatility,” *Economics Letters*, vol. 150, pp. 10–14, 2017.

[330] M. Katzfuss, D. Hammerling, and R. L. Smith, “A bayesian hierarchical model for climate change detection and attribution,” *Geophysical Research Letters*, vol. 44, p. 5720–5728, 2017.

[331] E. C. Massoud, H. Lee, P. B. Gibson, P. Loikith, and D. E. Waliser, “Bayesian model averaging of climate model projections constrained by precipitation observations over the contiguous United States,” *Journal of Hydrometeorology*, vol. 21, p. 2401–2418, 2020.