Sampling methods for solving Bayesian model updating problems
A tutorial

Lye, Adolphus; Cicirello, Alice; Patelli, Edoardo

DOI
10.1016/j.ymssp.2021.107760

Publication date
2021

Document Version
Final published version

Published in
Mechanical Systems and Signal Processing

Mechanical Systems and Signal Processing 159 (2021) 107760

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

journal homepage: www.elsevier.com/locate/ymssp

Sampling methods for solving Bayesian model updating problems: A tutorial

Adolphus Lye [a], Alice Cicirello [a,b], Edoardo Patelli [a,c,*]

[a] Institute for Risk and Uncertainty, University of Liverpool, United Kingdom
[b] Faculty of Civil Engineering and Geoscience, Delft University of Technology, Stevinweg 1, Delft, 2628 CN, Netherlands
[c] Centre for Intelligent Infrastructure, Department of Civil and Environmental Engineering, University of Strathclyde, United Kingdom

ARTICLE INFO

Article history:
Received 5 June 2020
Received in revised form 20 November 2020
Accepted 15 February 2021

Keywords:
Bayesian inference
Model updating
Markov Chain Monte Carlo
Transitional Markov Chain Monte Carlo
Sequential Monte Carlo
DLR-AIRMOD

ABSTRACT

This tutorial paper reviews the use of advanced Monte Carlo sampling methods in the context of Bayesian model updating for engineering applications. Markov Chain Monte Carlo, Transitional Markov Chain Monte Carlo, and Sequential Monte Carlo methods are introduced, applied to different case studies and finally their performance is compared. For each of these methods, numerical implementations and their settings are provided.

Three case studies with increased complexity and challenges are presented showing the advantages and limitations of each of the sampling techniques under review. 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.

The case studies presented in this paper consider the recorded data set as a single piece of information which is used to make inferences and estimations on time-invariant model parameters.

1. Introduction

In engineering design problems, mathematical models are used to investigate the virtual behaviour of structures under operational and extreme conditions. 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 [1]. The conventional model updating technique is the Finite Element model updating [2,3]. This approach is employed to perform point-estimates of physical parameters. Specifically, the input parameters of the physical model are "tuned" or updated by 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. 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. However, this type of approach faces three main problems: (i) it assumes that the mathematical model employed is able to capture the physics of the problem in full (i.e. not affected by

* Corresponding author at: University of Strathclyde, United Kingdom.
E-mail address: edoardo.patelli@strath.ac.uk (E. Patelli).

https://doi.org/10.1016/j.ymssp.2021.107760
0888-3270/

modelling errors and/or uncertainties); (ii) it does not readily take into account that the experimental data are usually affected by "noise" [4–7]; (iii) 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 [8–12] and not a single 'true' parameter value representation [13]. Broadly speaking, the approaches for model updating under uncertainty can be grouped into two categories: probabilistic and non-probabilistic. Among the probabilistic approaches, one of the most well-established is the Bayesian model updating framework developed by Beck and co-workers [14,15]. Within this framework, the physical parameters of the model to be updated are represented by probability density functions, 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 [16–22]. For real case applications, stochastic 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, making standard Monte Carlo method inapplicable.

This tutorial paper reviews and illustrates the use of three of the most popular advanced sampling techniques for approximate inference in the context of Bayesian model updating problems. Although the techniques are generally applicable, in this paper we will only focus on the inference of time-invariant parameters. Specifically, a simple introduction to the Markov Chain Monte Carlo (MCMC), Transitional Markov Chain Monte Carlo (TMCMC), and Sequential Monte Carlo (SMC) approaches is provided as well as guidelines and advises regarding the adoption of these techniques. These techniques are applied to three different case studies of increasing complexity to illustrate their application in engineering design problems as well as to assess their robustness, strengths and weaknesses. This tutorial is targeted at readers who may not be well-versed with Bayesian model updating and the advanced sampling techniques. The objective of this paper is to allow a much clearer understanding of the concept, differences, and the implementation of advanced sampling methods. For this purpose, each case study presented in this paper uses a data set which serves as a single piece of information that is used to estimate and update our knowledge on the time-invariant parameters. All the source codes, algorithms and examples used in this paper are made available to the readers as additional data and accessible also via GitHub:https://github.com/cossan-working-group/BayesianModelUpdating.

In Section 2 the problem of stochastic model updating is presented. Section 3 presents in details three sampling techniques (i.e. MCMC, TMCMC and SMC) and their underlying algorithms for Bayesian inference. In Section 4, three different case examples are presented, aimed at demonstrating and comparing the effectiveness of each of the advanced sampling techniques under different set-ups. The three case examples which will be discussed in this paper are the following: spring-mass system under a static load, 2-dimensional inverse eigenvalue problem, and the 18-dimensional non-linear DLR-AIRMOD model updating problem. For each problem discussed, an evaluation and discussion of the key strengths and limitations of each of the advanced sampling methods will be provided before finally drawing the paper to a conclusion in Section 6.

2. Stochastic 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 [23].

The mathematical relation between the quantity of interest to be assessed D (e.g. the frequency response function of the system) and the model prediction $M(\boldsymbol{x}, \boldsymbol{\theta})$ can be expressed in general as [24]:

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

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 by using a probability density function. By doing so, the modeller can construct a stochastic model (or a class of models [14])) to predict probabilistically 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 [25–27] or in general by Monte Carlo simulation approaches [28–30]. In the latter approach, the statistics of D are obtained by first generating n realisations of the $\boldsymbol{\theta}_k$ (for $k = 1, \ldots, n$) parameters from a known joint probability density function $\pi(\boldsymbol{\theta})$. Then, for each realization of $\boldsymbol{\theta}_k$ the model is evaluated to obtain the corresponding realization of D_k (i.e. $D_k = M(\boldsymbol{x}, \boldsymbol{\theta}_k)$). By repeating this process for all n samples, we eventually obtain a sample distribution of $\hat{\boldsymbol{D}}$ [31,32]. There are three advantages of adopting the Monte Carlo approach: Firstly, the Monte Carlo technique is applicable to any problem including non-smooth or non-linear cases (see e.g. [33,34]). Secondly, 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 thirdly, the computation performed by the Monte Carlo technique is easily parallelisable [34].

Therefore, 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 [35]. There are two main statistical approaches to

identify the parameters of a statistical model given a set of observations: the Bayesian model updating and the frequentist approach. The Maximum Likelihood Estimator (MLE) is one of the most often used estimator in the frequentist literature [36–39] by finding the parameter values that maximize the likelihood of observing the $\boldsymbol{D}$ given the parameters $\boldsymbol{\theta}$). The Bayesian approach casts this inverse problem as a Bayesian Inference problem [14,15,40] and it is explained in the following subsection.

Since the focus of this paper is to make inferences on $\boldsymbol{\theta}$, the representation of the model output $M_k(\boldsymbol{x}, \boldsymbol{\theta})$, the latter can be simplified as $M(\boldsymbol{\theta})$. It should be noted that the model $M(\boldsymbol{\theta})$, relating $\boldsymbol{\theta}$ and D_k, can be either linear or non-linear.

2.1. 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.

In particular, when a set of n independent and identically distributed observations $D_1, D_2, \ldots, D_n$ become available, the prior belief is updated using Bayes' theorem [41] leading to the posterior distribution of the parameter of interest:

$$P(\boldsymbol{\theta}|\boldsymbol{D}) = \frac{P(\mathbf{D}|\boldsymbol{\theta}) \cdot P(\boldsymbol{\theta})}{P(\mathbf{D})}$$

whereby

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

2.1.1. Prior distribution

The prior distribution, $P(\boldsymbol{\theta})$, 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 [42]. In theory, any type of prior distribution can be used depending on the amount of information available [40]. However, in practice Uniform and Normal distributions are the most common types of priors adopted.

If what is known about the parameter(s) is/are its upper and lower bounds, then a Uniform distribution could be used as the non-informative prior distribution based on the principle of Maximum Entropy [43,44]. 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 [45]; structural parameters of a composite structure [46]; crack parameters of a beam structure [47]; stiffness and mass parameters of a DLR-AIRMOD structure [48]; and stiffness parameters of a cantilever beam [49].

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 [50]; joint-stiffness parameters of the stochastic model for a joint contact surface [51]; the logarithmic ground truth system parameters of a three degrees-of-freedom system [52]; the cosmological parameters used in a supernovae analysis [53]; and state parameters of the dynamical model used for real-time defect detection of high-speed train wheels [54].

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

$$P(\boldsymbol{\theta}) = P\left(\theta^1, \ldots, \theta^{N_d}\right) = \prod_{d=1}^{N_d} P\left(\theta^d\right)$$

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

2.1.2. Likelihood function

The likelihood function, $P(\boldsymbol{D}|\boldsymbol{\theta})$, reflects the degree of agreement between the obtained measurements, $\boldsymbol{D}$, and the output obtained from the mathematical model $M(\boldsymbol{\theta})$ used to physically describe the system. For the case studies presented in this paper, we assume only one model 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 [55–58]). It needs to be pointed out that the model output $M(\boldsymbol{\theta})$ considered in this paper is purely deterministic. For this reason, the likelihood function has to capture the degree of agreement between $\boldsymbol{D}$ and $M(\boldsymbol{\theta})$ for each set of possible $\boldsymbol{\theta}$ values. As

such, the likelihood function is a function of $\boldsymbol{\theta}$ and not of $\boldsymbol{D}$. Assuming that the measurements D_k (for $k = 1, \ldots, n$) are independently, identically distributed, the likelihood function takes on the following mathematical form:

$$P(\boldsymbol{D}|\boldsymbol{\theta}) = \prod_{k=1}^{n} P(D_k|\boldsymbol{\theta})$$

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 [59–70]:

$$P(\boldsymbol{D}|\boldsymbol{\theta}) = \left(\prod_{k=1}^{n} \frac{1}{\sigma_k \cdot \sqrt{2 \cdot \pi}}\right) \cdot exp\left[-\sum_{k=1}^{n} \frac{(D_k - M(\boldsymbol{\theta}))^2}{2 \cdot \sigma_k^2}\right]$$

As seen in Eq. (1), the expression $(D_k - M_k(\boldsymbol{\theta}))$ simply yields error ϵ_k between the k^{th} measurement and the model output while σ_k^2 is a hyper-parameter that can be interpreted as the variance of ϵ_k. For cases whereby correlations are present between the measurements D_k, 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:

$$P(\boldsymbol{D}|\boldsymbol{\theta}) = \frac{1}{\sqrt{|\Sigma| \cdot (2 \cdot \pi)^n}} \cdot exp\left[-\frac{1}{2} \cdot \sum_{k=1}^{n}\sum_{l=1}^{n} (D_k - M(\boldsymbol{\theta})) \cdot \boldsymbol{\Sigma}^{-1} \cdot (D_l - M(\boldsymbol{\theta}))\right]$$

There are two possible ways to decide on the value of σ_k. The first would be through estimation via the mean squared error of $(D_k - M(\boldsymbol{\theta}))$. The second would be to set it as a fixed parameter based on prior calculations or knowledge. The strategy commonly adopted is to set σ_k to correspond to the standard deviation of ϵ_k, especially if the latter follows a prescribed distribution. For instance, a common choice for ϵ_k is a zero-mean Normal distribution with standard deviation σ_ϵ, then $\sigma_k = \sigma_\epsilon$. This strategy will be used in the tutorials to justify the choice of σ_k for the likelihood function.

It is worth noting that different forms of likelihood functions have been adopted in literature (see e.g. [71–74,65,75,76]) to capture the degree of agreement between $\boldsymbol{D}$ and $M(\boldsymbol{\theta})$, as summarised in Table 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 1 are compared in Fig. 1 for the case of a mono-dimensional θ. Note that the plot for the lognormal likelihood function is not included in the Fig. 1 as it is defined in the logarithmic space. Its shape profile, however, follows that of a Normal distribution.

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

$$log(\boldsymbol{D}|\boldsymbol{\theta})) = \sum_{k=1}^{n} log(P(D_k|\boldsymbol{\theta}))$$

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_k as shown in Eq. (3). Another advantage of using the log-likelihood is that the addition operation is much faster to compute than the product operation.

For this paper, the case studies presented are limited to 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 [77]. In addition, there are instances whereby the model output itself can be stochastic. Under such circumstances, one approach would be to simply adopt the use of Kernel densities to estimate the likelihood function

Table 1
Examples of typical likelihood functions used for model updating.

Type	Likelihood function, $P(D_k\|\boldsymbol{\theta})$	Reference
Normal Distribution	$\left(\prod_{k=1}^{n} \frac{1}{\sigma_k \cdot \sqrt{2 \cdot \pi}}\right) \cdot exp\left[-\sum_{k=1}^{n} \frac{(D_k - M(\boldsymbol{\theta}))^2}{2 \cdot \sigma_k^2}\right]$	[59–70]
Lognormal Distribution	$\frac{1}{\sqrt{2\pi \cdot log\left(1 + \frac{\sigma_k^2}{M(\boldsymbol{\theta})^2}\right)} \cdot D_k} \cdot exp\left[-\frac{(log(D_k) - log(M(\boldsymbol{\theta})))^2}{2 \cdot log\left(1 + \frac{\sigma_k^2}{M(\boldsymbol{\theta})^2}\right)}\right]$	[73,74,65,75,76]
Inverse Error	$1 - exp\left[-\sqrt{\frac{1}{(D_k - M(\boldsymbol{\theta}))^2}}\right]$	[71]
Inverse Squared Error	$1 - exp\left[-\frac{1}{(D_k - M(\boldsymbol{\theta}))^2}\right]$	[71]
Exponential Distribution	$\frac{1}{2 \cdot \sigma_k^2} \cdot exp\left[-\frac{\frac{1}{n}\sum_{k=1}^{n} (D_k - M(\boldsymbol{\theta}))^2}{2 \cdot \sigma_k^2}\right]$	[72]
Truncated Normal Distribution	$\frac{\sqrt{2}}{\sqrt{\pi} \cdot \sigma_k} \cdot exp\left[-\frac{\frac{1}{n}\sum_{k=1}^{n} (D_k - M(\boldsymbol{\theta}))^2}{2 \cdot \sigma_k^2}\right]$	[72]

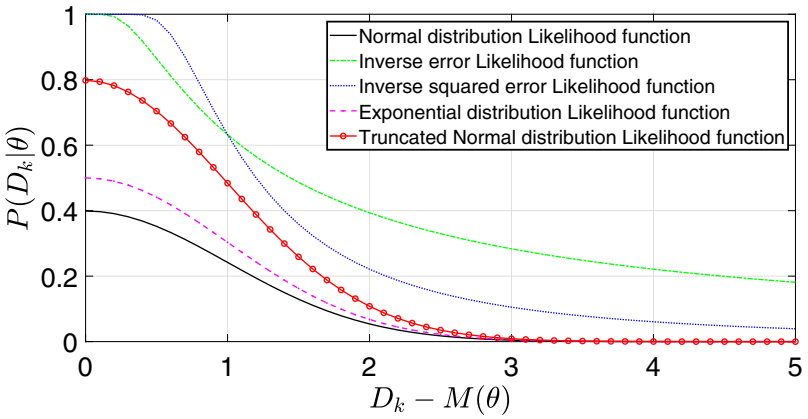


Fig. 1. Comparison of different likelihood functions for the case of a mono-dimensional θ: Normal distribution (black solid line), Inverse Error (green dashed-dotted line), the Inverse Squared Error (blue dotted line), exponential (magenta dashed line), and Truncated normal (red solid line with circles). For all the plots, σ_k is set as 1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

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 [77]. Alternatively, one could also turn to the use of approximate likelihood functions and adopt the technique of Approximate Bayesian Computation (ABC) [78,79]. These approximate likelihood functions capture the discrepancy between the $\boldsymbol{D}$ and the model outputs using stochastic distance metrics such as: Euclidian [80]; Mahalanobis [81]; and Bhattacharyya distances [82]. More details to ABC and the various stochastic distance metrics can be found in the respective references.

2.1.3. Evidence function

The evidence function, $P(\boldsymbol{D})$, serves as a normalizing constant of the Bayesian formula to ensure that the posterior (see Eq. (2)) integrates to 1. In Bayesian inference, the probability of the observation $P(\boldsymbol{D})$ 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, we obtain the resulting proportionality relation for the posterior distribution up to a normalising constant [83]:

$$P(\boldsymbol{\theta}|\boldsymbol{D}) \propto P(\boldsymbol{D}|\boldsymbol{\theta}) \cdot P(\boldsymbol{\theta})$$

Based on Eq. (8), the form of the posterior distribution is only known implicitly.

2.1.4. Posterior distribution and its estimation

The posterior distribution, $P(\boldsymbol{\theta}|\boldsymbol{D})$, 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 [84–88], 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 multimodal. In these situations it would be computationally expensive even if we limit our interest to the evaluation of the mean and variance of the posterior distribution analytically.

The generally applicable numerical technique for estimating distributions is the Monte Carlo method [89]. In particular, Monte Carlo Methods can be used to efficiently evaluate the statistics of $\boldsymbol{\theta}$, rather than the full posterior distribution. Suppose we are able to generate samples from $P(\boldsymbol{\theta}|\boldsymbol{D})$, it is possible to estimate the moments of the posterior distribution as follows [90]:

$$E[\boldsymbol{\theta}] = \sum_{i=1}^{N} \boldsymbol{\theta}_i \cdot P(\boldsymbol{\theta}_i|\boldsymbol{D})$$

$$E\left[\boldsymbol{\theta}^2\right] = \sum_{i=1}^{N} \boldsymbol{\theta}_i^2 \cdot P(\boldsymbol{\theta}_i|\boldsymbol{D})$$

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

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

whereby $E[\boldsymbol{\theta}]$ and $E\left[\boldsymbol{\theta}^2\right]$ are obtained from Eqs. (9) and (10) respectively. Marginals and quantiles of the distribution can also be computed using the same realisations.

However, obtaining independent samples from the posterior is difficult using only the standard Monte Carlo approach. This is because the posterior distribution is known only implicitly (i.e. point-wise values of the distribution are known only after evaluating the prior and likelihood function). Therefore it is not possible to generate samples directly for the posterior. Instead, Markov Chain Monte Carlo (MCMC) 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 $P(\boldsymbol{\theta}|\boldsymbol{D})$ [91]. MCMC does not require the evaluation of the evidence, and Eq. (8) can be used directly. MCMC only requires evaluation of the joint distribution of Eq. (8) up to a proportionality factor and point-wise for any generated sampled of $\boldsymbol{\theta}$.

Therefore, MCMC algorithms return samples $\boldsymbol{\theta}_i, i = 1, \ldots, N$ where each sample can be assumed to be drawn from $P(\boldsymbol{\theta}|\boldsymbol{D})$. Different advanced sampling methods to generate samples of the posterior distribution have been proposed and they will be reviewed in this paper.

2.1.5. 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 [92]; 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 [49]; in structural health monitoring by identifying the position and severity of a crack in a suspension arm of a car [71]; to update the material dependent constants of the Paris-Erdogan Law used to predict crack growth rate in a carbon-steel Nuclear piping [93]; to perform on-line estimation of parameters of building energy models based on information from in situ sensor [94]; and to estimate the most probable leakage scenarios for the purpose of leakage detection in water distribution networks [95]. Details to the Bayesian model updating set-up in these references are summarised in Table 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 [96]. This, however, will not be discussed given that it involves the use of structural reliability methods which is beyond the scope of this paper.

3. Review of advanced Monte Carlo samplers

3.1. Markov Chain Monte Carlo

The MCMC sampler is a sampling technique introduced by Metropolis [97] which encompasses two 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 $i = 1, 2, \ldots, N$ whereby the value of θ_{i+1} depends only on the previous value θ_i [98]. This is also known as the Markov property [99]. 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 [100] which is determined by a transition probability distribution function. In a time-homogeneous Markov process, the distribution of the generated samples θ_i would converge to a stationary distribution whereby the distribution of the samples becomes stable. However, the initial samples of the Markov chain are in general not distributed according to the stationary distribution and are thus not representative of the stationary distribution. Thus, it becomes a practice to discard the initial $N_{burn-in}$ number of samples. This is known as the burn-in and $N_{burn-in}$ corresponds to the burn-in length of the Markov chain [101]. In the context of Bayesian model updating, this stationary distribution corresponds to the posterior distribution.

There are many variants of MCMC techniques which are currently in existence and two of the most commonly used variants are the Metropolis–Hastings (MH) sampler [102] and the Gibbs sampler [103,104]. A problem with the Gibbs sampler,

Table 2
Summary of Bayesian Model Updating set-up and sampling technique employed.

Application	Prior	Likelihood(s)	Sampling technique
Estimate torsional stiffness parameter for a cantilever beam [49]	Uniform	Normal	MCMC
Identify the material dependent constants of the Paris-Erdogan Law for crack growth rate prediction [93]	Normal	Normal	MCMC
Quantify the discrete element methods prediction of granular materials' behaviour [92]	Uniform	Normal	TMCMC
Leakage detection in water distribution networks [95]	Uniform	Normal	TMCMC
Online monitoring (crack detection) in a suspension arm of a car [71]	Uniform	Normal, Inverse Error, Inverse Squared Error	TMCMC
On-line parameter estimation of building energy models [94]	Normal	Normal	SMC

however, is in the selection of an appropriate conditional probability distribution to represent the posterior distribution [105]. In general, this may not be trivial because the posterior may be functionally complex. This makes the implementation of the Gibbs sampler less general and in this regard, the MH variant of MCMC will be adopted to address the problems presented in this paper. One key strength of the MH algorithm which motivated its use, 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. the posterior distribution in the form of Eq. (8)) is known and that the values of that function can be computed [102]. Without the loss of generality, this section will first elaborate the steps of the MH algorithm for sampling from a mono-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 so-called 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 posterior is defined. In addition, the common criteria in deciding $q(\theta^*|\theta_i)$ is that it has to be symmetric. This makes the Normal and Uniform distributions [106] the main options for $q(\theta^*|\theta_i)$ although it has also been argued in [107] 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. Fig. 2 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:

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

whereby $P(\theta^*|\boldsymbol{D})$ represents the posterior value evaluated at the candidate sample θ^*, and $P(\theta_i|\boldsymbol{D})$ 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. (12) with its definition from Eq. (8), we obtain:

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

From Eq. (13), it can be seen that the normalisation constant $P(\boldsymbol{D})$ is cancelled out. This further justifies why there is no need to evaluate $P(\boldsymbol{D})$ and that the computation of Eq. (12) can be done using an unnormalised posterior (see Eq. (8)). This allows the MH algorithm to perform sampling on such distributions. For $q(\theta^*|\theta_i)$, a symmetrical function (e.g. Normal or Uniform distribution) that is centered about θ_i is usually considered such that $q(\theta^*|\theta_i) = q(\theta_i|\theta^*)$. As a result, the acceptance probability, α, in Eq. (12) becomes simplified:

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

What Eq. (14) implies is that the candidate sample θ^* is always accepted if the samples are moving towards the region of high probability density (i.e. $\frac{P(\theta^*|\boldsymbol{D})}{P(\theta_i|\boldsymbol{D})} > 1$), 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 $\alpha \geqslant r$, the proposed sample θ^* is accepted (i.e. $\theta_{i+1} = \theta^*$). Otherwise, θ^* is rejected (i.e. $\theta_{i+1} = \theta_i$).

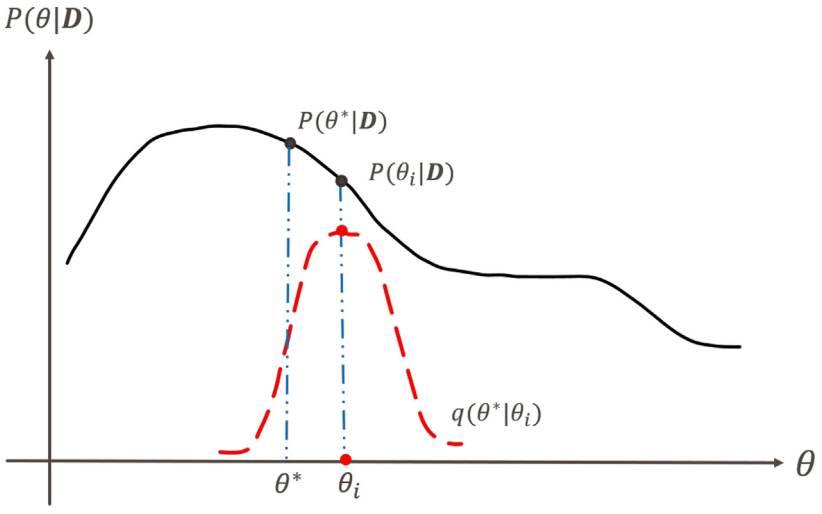


Fig. 2. Metropolis-Hasting sampling - the proposal distribution $q(\theta^*|\theta_i)$ (red dotted curve); posterior $P(\theta|D)$ (black solid curve); current sample θ_i; and proposed sample θ^*. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The characteristic 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 parallelized. In fact, there have been developments made in recent years to achieve this as seen in the works by Wilkinson (2005) [108] and Brockwell (2006) [109] though efforts have proved to be non-trivial. Besides the MH algorithm, there are also various other MCMC algorithms implementation available including Slice sampling [110], Hamiltonian Monte Carlo methods [111,112], Metropolis-adjusted Langevin algorithm [113–115,112], Multiple-try Metropolis [116,117], Reversible-jump MCMC [118], and the Pseudo-marginal Metropolis–Hastings algorithm [119]. Detailed explanations to each of these algorithms can be found in their respective references.

In the tutorial, 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. Should the value be too small, it results in a small jump-size between a sample and the next successive sample of the chain. This leads to a high serial dependence between successive samples, giving rise to high autocorrelation between these samples [120]. On the other hand, when the tuning parameter is too large, it may result in many of the proposed samples lying outside the range of the posterior and the entire sampling space not being sampled efficiently. This gives rise to a high rejection rate of the proposed samples, thus, making sampling process inefficient and ineffective. Under specific condition, the optimal value of σ_p is the one which produces an acceptance rate around 0.234 [121]. However, for practical applications, the usual 0.234 might be inefficient even with seemingly regular targets [122]. Intuitively, the acceptance ratio is a trade-off between making too many small accepted steps and making too many large proposals that get rejected. In fact, for any value of acceptance rate between 0.15 to 0.50, the efficiency of the algorithm is still at least 80% [120].

To illustrate the effects of the tuning parameter on the sampling process, we allow the MH sampler to generate 1000 samples from a posterior defined by a Uniform prior ranging between 1.0 to 1000, and a Normal likelihood function with standard deviation 1.0. The model used in this example is a simple linear model in the form of: $M(\theta) = \theta \cdot x$ whereby $M(\theta)$ is the model output, θ is the uncertain model parameter we wish to estimate from the posterior, and x is the model input. The proposal distribution is a Normal distribution with standard deviation σ_p. For this example, 3 different values of σ_p is used: $100.5, 22.5$, and 1.5. The resulting trace plots of the generated samples and their corresponding acceptance rates are presented in Fig. 3. In the literature, the efficiency of the MCMC sampler can be interpreted as the number of iterations required by the sampler to attain the required degree of accuracy and precision of the estimate [120]. To avoid the need for the accept/reject step as well as the need for tuning, one could turn to tune-free MCMC algorithms such as Gibbs sampling and Slice sampling.

The burn-in length of the chain must be checked and one way to do so would be through constructing a trace plot and identifying the sample number at which the plot begins to converge [101]. As an illustration, we will be using the same set-up as the earlier example used to generate Fig. 3. 1000 samples will be generated via MH sampling, with 0 burn-in length. The starting value of the chain is randomly sampled from the Uniform prior and is set as $\theta_1 = 444$. 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 Fig. 4. From the trace plot in Fig. 4(a), it can be observed that the plot starts to converge after 40 samples are obtained indicating that $N_{burn-in} = 40$. Fig. 5 illustrates the resulting trace plot and histogram profile after accounting for burn-in.

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

1. Set $i = 1$; sample $\theta_i \sim P(\theta)$.
2. Generate candidate sample $\theta^* \sim q(\theta^*|\theta_i)$.
3. Evaluate the posterior distribution at the proposed sample (i.e. $P(\theta^*|\boldsymbol{D})$).
4. Compute the acceptance ratio, α, from Eq. (14).
5. Sample $r \sim U(0, 1]$. If $\alpha \geqslant r$, set $\theta_{i+1} = \theta^*$. Otherwise, set $\theta_{i+1} = \theta_i$.

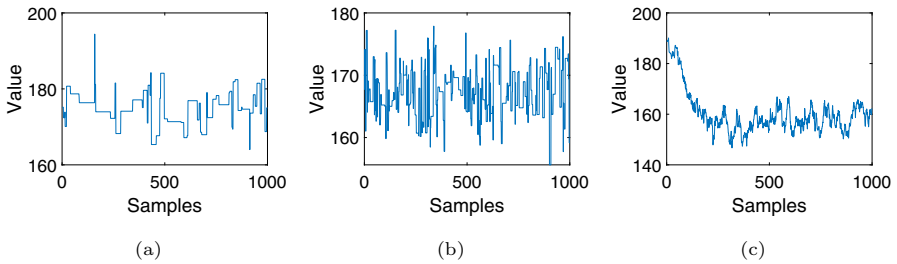


Fig. 3. The corresponding trace plots obtained with 1000 samples using: (a) $\sigma_p = 100.5$ giving an acceptance level of 0.061; (b) $\sigma_p = 22.5$ giving an acceptance level of 0.234; (c) $\sigma_p = 1.5$ giving an acceptance level of 0.865.

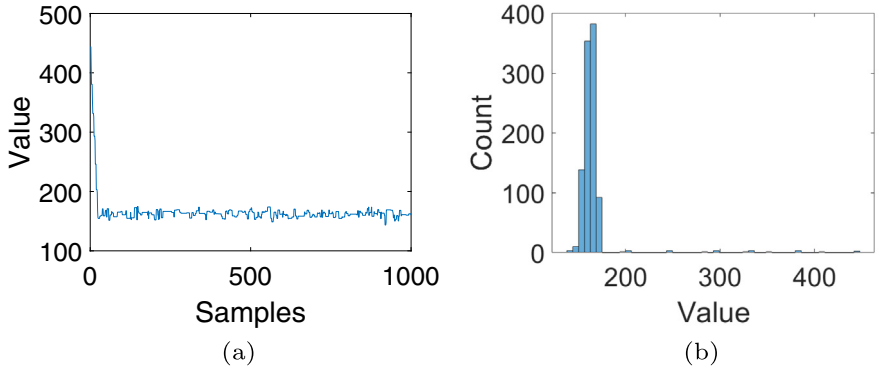


Fig. 4. Example of the resulting trace plot (a) and histogram (b) obtained from MH sampling of 1000 samples with 0 burn-in.

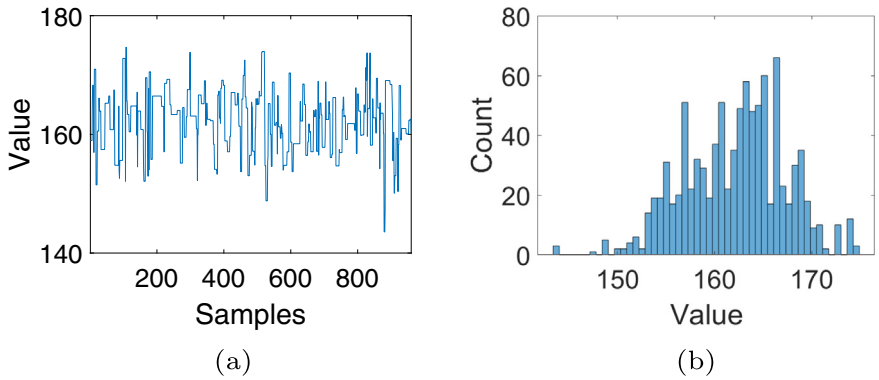


Fig. 5. The resulting trace plot (a) and histogram (b) obtained from MH sampling of 1000 samples after discarding the initial $N_{burn-in} = 40$ samples.

6. Set $i = i + 1$ and repeat steps (2) to (6) until termination criteria is met (e.g. 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. Two well-established approaches can be adopted to sample from multi-variate posteriors: block-wise and 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 [123,124]. In essence, variables across all dimensions are updated simultaneously rather than sequentially as per the case in component-wise approach. A key problem with this approach is that the acceptance rate drops with 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 [125–127]. The proposal distribution can be uni-variate or multi-variate, the latter taking the form $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}_i) = \prod_{d=1}^{N_d} q\left(\theta^{d*}|\theta_i^d\right)$ 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 [128]. In this paper, the component-wise approach is adopted to sample from a multi-variate posterior and a multi-variate Normal distribution would be used as the choice for $q(\boldsymbol{\theta})$ with covariance matrix $\boldsymbol{\Sigma}_p$ which now serves as the tuning parameter of the sampler. $\boldsymbol{\Sigma}_p$ 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, we denote $\boldsymbol{\theta}_i^d$ (for $d = 1, \ldots, N_d$) 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, $\boldsymbol{\theta}_i^d = \left\{\theta_{i+1}^1, \ldots, \theta_{i+1}^{d-1}, \theta_i^d, \theta_i^{d+1}, .., \theta_i^{N_d}\right\}$ while $\boldsymbol{\theta}^{d*} = \left\{\theta_{i+1}^1, \ldots, \theta_{i+1}^{d-1}, \theta^{d*}, \theta_i^{d+1}, .., \theta_i^{N_d}\right\}$. Using this generalised convention, the algorithmic description of the MH sampler in sampling from a general N_d-dimensional posterior is presented in Algorithm 1.

Algorithm 1 Component-wise MH algorithm (N_d-dimensional case)

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

The MH algorithm has been implemented in numerous areas of research. For instance, to predict precipitation behaviours in Nickel-Titanium alloys via Bayesian probability [129]; to analyse an electrochemical impedance spectra and estimate the conductivity of a Lithium ion within a solid-state oxide electrolyte [130]; to predict and quantify the uncertainty associated with the forecasts for daily river flow rate of Zhujiachuan River [131]; and to sample classical thermal states from one-dimensional Bose–Einstein quasi-condensates under the classical fields approximation [132]; to update the finite element model of a concrete structure [133]; to quantify the uncertainty associated with the joint model parameters of a stochastic generic joint model [51]; to perform joint input-state-parameter estimation for wave loading [52]; to perform Bayesian system identification of dynamical systems [134]; and to perform Bayesian model identification of higher-order frequency response functions of structures [45].

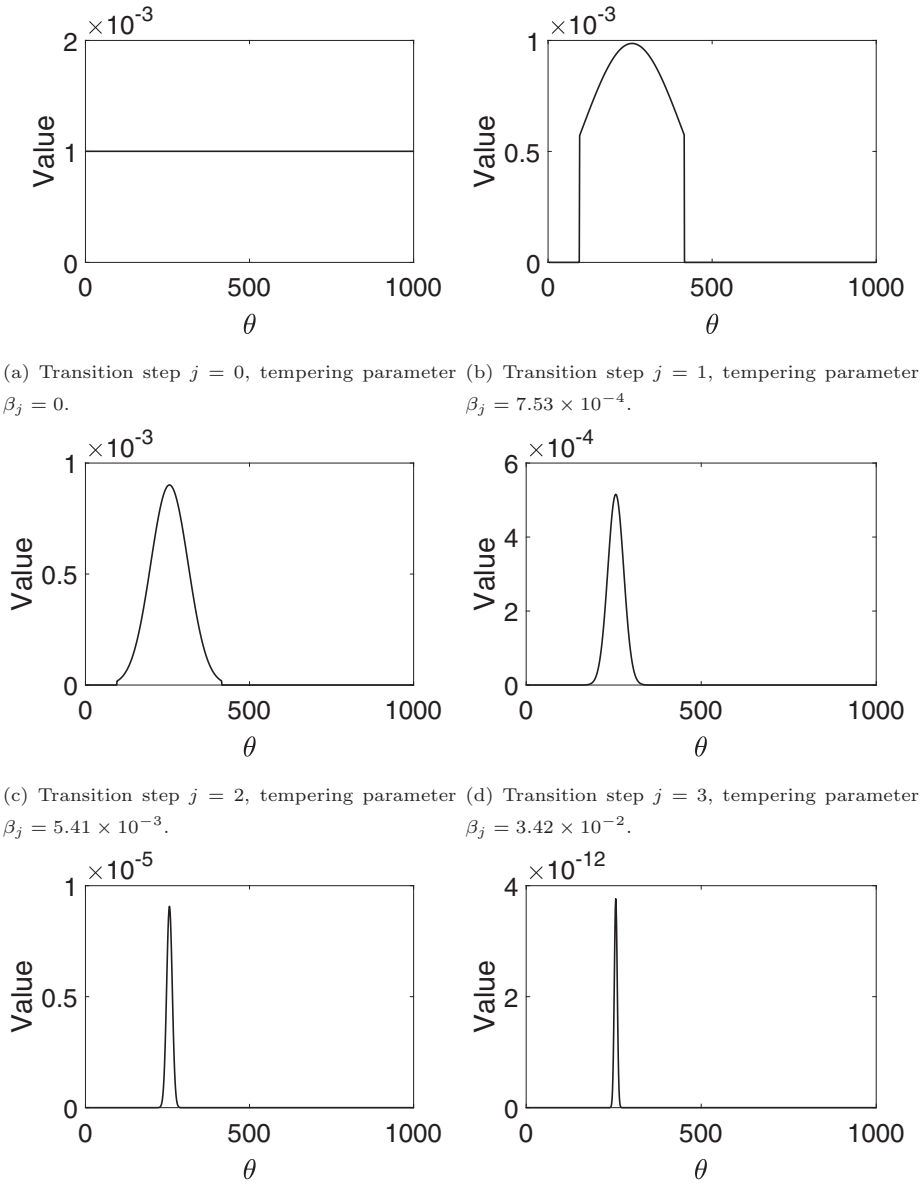
3.2. Transitional Markov Chain Monte Carlo

The TMCMC sampler is based on the adaptive Metropolis–Hastings technique [135] 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 MCMC and the TMCMC technique is that MCMC samples are obtained through one or few, successive (very) long Markov Chains of length N, whereas TMCMC samples are obtained through N independent single-step 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 [55]:

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

Here, j denotes the transition step number taking values between 0 to m, where m denotes the last iteration number. β_j is the tempering parameter which takes values such that $\beta_0 = 0 < \beta_1 < \cdots < \beta_{m-1} < \beta_m = 1$. This allows for the transitional distribution to transit from the prior to the posterior distribution (i.e. $P^0 = P(\boldsymbol{\theta})$ to $P^m = P(\boldsymbol{\theta}|\boldsymbol{D})$). As an illustration, Fig. 6 depicts a series of analytical plots show the evolution of a mono-dimensional transitional distribution from a Uniform prior (i.e. $P(\theta) \sim U[1, 1000]$) to a sharp-peaked Normal posterior (i.e. $P(\theta|\boldsymbol{D}) \sim N\left(252.26, 1.17^2\right)$) through the use of a Normal likelihood as shown in Eq. (5) with parameter $\sigma_k = 1$ and model $M(\theta) = \theta \cdot x$. Here, θ is the estimated parameter and x is the model input.

The important aspect of the TMCMC sampler is the determination of 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 $P(\boldsymbol{D}|\boldsymbol{\theta})^{\beta_{j-1}}$ to $P(\boldsymbol{D}|\boldsymbol{\theta})^{\beta_j}$ 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, we use the same set-up used for Fig. 6. The value of $\Delta\beta_j$ is varied and the corresponding value of acceptance



(a) Transition step $j = 0$, tempering parameter $\beta_j = 0$.

(b) Transition step $j = 1$, tempering parameter $\beta_j = 7.53 \times 10^{-4}$.

(c) Transition step $j = 2$, tempering parameter $\beta_j = 5.41 \times 10^{-3}$.

(d) Transition step $j = 3$, tempering parameter $\beta_j = 3.42 \times 10^{-2}$.

(e) Transition step $j = 4$, tempering parameter $\beta_j = 2.42 \times 10^{-1}$.

(f) Transition iteration $j = 5$, tempering parameter $\beta_j = 1$.

Fig. 6. Evolution of the transition distribution from an initial Uniform prior distribution (subplot a) to the final posterior distribution (subplot f) through the use of a Normal likelihood.

rate will be obtained and shown in Fig. 7. From Fig. 7, it is observed that the value of acceptance rate drops from 1 when $\Delta\beta_j = 0$ to approximately 0.18 when $\Delta\beta_j = 1$. This is due to the large difference in the shape function between the prior and the posterior when $\Delta\beta_j$ 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 $\Delta\beta_j = 0$, the acceptance rate is 1 given that the samples are generated from the same distribution, leading to a 100% acceptance rate. This brings a 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 $P(\boldsymbol{D}|\boldsymbol{\theta}_i)^{\Delta\beta_j}$ as close to 100% as possible [55]. For a mono-dimensional case, the COV of $P(\boldsymbol{D}|\theta_i)^{\Delta\beta_j}$ is defined as:

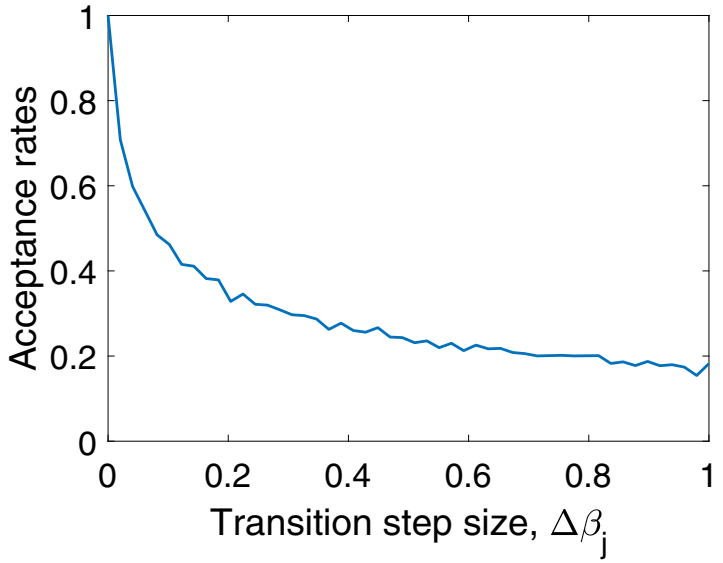


Fig. 7. Plot of acceptance rates against the transition step size $\Delta\beta_j$ from the prior based on the same set-up used for Fig. 6.

$$COV(\beta_j) = \frac{\sigma\left(P(\boldsymbol{D}|\theta_i)^{\Delta\beta_j}\right)}{\mu\left(P(\boldsymbol{D}|\theta_i)^{\Delta\beta_j}\right)}$$

whereby $\sigma\left(P(\boldsymbol{D}|\theta_i)^{\Delta\beta_j}\right)$ and $\mu\left(P(\boldsymbol{D}|\theta_i)^{\Delta\beta_j}\right)$ are the standard deviation and mean of the value set $P(\boldsymbol{D}|\theta_i)^{\Delta\beta_j}$ (for $i = 1, \ldots, N$). Here, $COV(\beta_j)$ is a function of β_j for a given known value of β_{j-1}. For a multi-dimensional case, $\sigma\left(P(\boldsymbol{D}|\theta_i)^{\Delta\beta_j}\right)$ and $\mu\left(P(\boldsymbol{D}|\theta_i)^{\Delta\beta_j}\right)$ are the standard deviation and mean of the value set $P(\boldsymbol{D}|\boldsymbol{\theta}_i)^{\Delta\beta_j}$ respectively whereby $P(\boldsymbol{D}|\boldsymbol{\theta}_i)$ is simply the likelihood evaluated at sample set $\boldsymbol{\theta}_i = \left\{\theta_i^1, \theta_i^2, \ldots, \theta_i^{N_d-1}, \theta_i^{N_d}\right\}$. Here, N_d is the total number of dimensions of $\boldsymbol{\theta}_i$. After obtaining $COV(\beta_j), \beta_j$ can then be determined analytically from β_{j-1} using the argument of the minimum of the absolute difference between $COV(\beta_j)$ and 1 (i.e. 100%) as shown in Eq. (17) [55,136]:

$$\beta_j = argmin_{\beta_j}\left\{|COV(\beta_j) - 1|\right\}$$

Once β_j is calculated, the transition distribution P^j can then be determined using Eq. (15).

To provide an understanding of the workings behind the TMCMC sampler, we will first explain its procedure in sampling from a mono-dimensional posterior. In practice, at transition step $j = 0$ (i.e. $\beta_j = 0$), N samples are generated from the prior via direct random sampling using standard Monte Carlo method. This can be done since the prior is usually a well-defined distribution. For transition steps $j \geqslant 1$ (i.e. while $\beta_j < 1$), β_j is first calculated from β_{j-1} using Eq. (17). From there, the transitional distribution P^j is defined using Eq. (15) and N samples are then obtained from P^j using MH sampler through the following procedure: First, a statistical weight function $\hat{w}(\theta_i)$ is determined to describe the statistical (or importance) weight associated with each sample θ_i in the current iteration. This statistical weight function $\hat{w}(\theta_i)$ is mathematically defined in Eq. (18) as:

$$\hat{w}(\theta_i) = \frac{P(D|\theta_i)^{\Delta\beta_j}}{\sum_{i=1}^{N} P(D|\theta_i)^{\Delta\beta_j}}$$

θ_i (for $i = 1, \ldots, N$) is then resampled with replacement, according to $\hat{w}(\theta_i)$, from the sample set obtained in the previous transition step. This process is analogous weighted random sampling [137] and an illustration to this process is provided in Fig. 8. Each of the resampled θ_i is then set as the starting sample for the single-step Markov chain. This creates N single-step Markov chains whereby each chain generates 1 sample. For each of the N chains, a candidate sample θ_i^* is generated from a Normal proposal distribution $q(\theta_i^*|\theta_i)$ with mean θ_i and covariance matrix $\boldsymbol{\Sigma}$ [55]. The covariance matrix also serves as the tuning parameter of the MH sampler and is mathematically defined in Eq. (19):

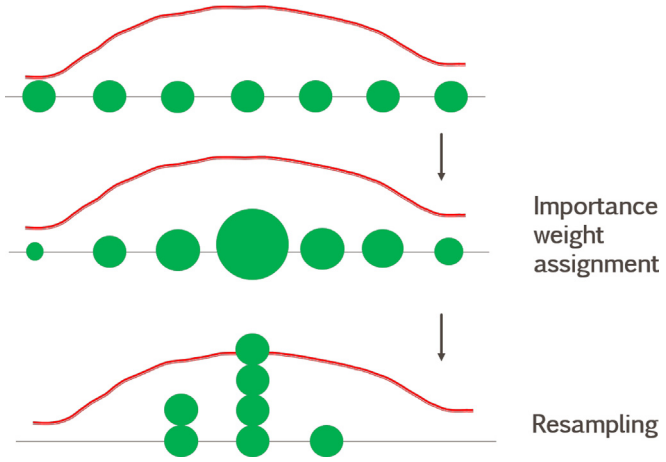


Fig. 8. Illustration as to how the resampling procedure is done according to the statistical weights of the samples in green. The red curve here represents the statistical weight function, $\hat{w}(\theta_i)$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$\boldsymbol{\Sigma} = \gamma^2 \sum_{i=1}^{N} \hat{\boldsymbol{w}}(\theta_i) \cdot \left[\{\theta_i - \bar{\theta}\} \times \{\theta_i - \bar{\theta}\}^T \right]$$

whereby

$$\bar{\theta} = \sum_{i=1}^{N} \theta_i \cdot \hat{\boldsymbol{w}}(\theta_i)$$

Here, $\bar{\theta}$ denotes the mean value of the sample set θ_i in the current iteration j, and γ is the scaling parameter of $\boldsymbol{\Sigma}$ whose optimum value was determined to be 0.2 [55]. Finally, the samples are updated by accepting or rejecting candidate samples θ_i^* using Algorithm 1. When done, the algorithm proceeds to recompute β_{j+1} and the transitional distribution P^{j+1} for iteration $j = j + 1$. This entire process is repeated until when $\beta_j = 1$.

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. For the purpose of illustration, we refer to the same set-up used in Fig. 6. Two values of transition step size from the Uniform prior are used for this example, $\Delta\beta_j = 0.000753$ (optimised step size) and $\Delta\beta_j = 0.0204$ (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 Fig. 9. With $\Delta\beta_j = 0.000753$, Fig. 9(a) depicts 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 324 out of 1000 (i.e. 32.4% of samples), having such weight values. The reason for this is due to the resulting transition distribution now

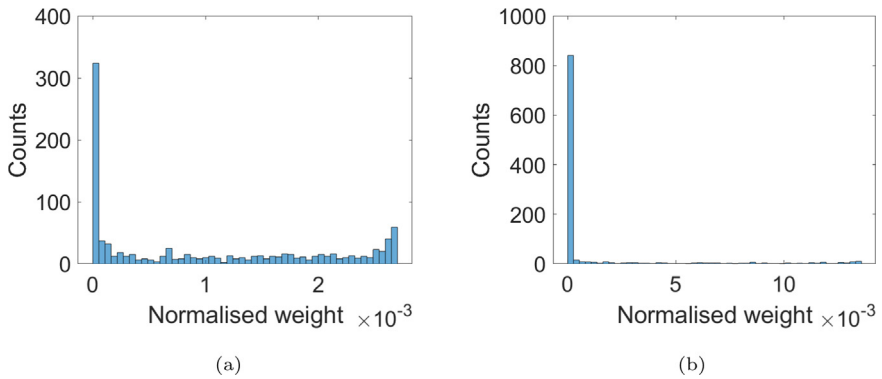


Fig. 9. Normalised weight distribution when $\Delta\beta_j = 0.000753$ (subplot a) and when $\Delta\beta_j = 0.0204$ (subplot b). The transition step $\Delta\beta_j$ is with respect to the Uniform prior in Fig. 6.

being defined over a significantly smaller sample space compared to the Uniform prior (see Fig. 6(a) and (b)) leading to smaller weights assigned to these samples which now lie in the region of low probability defined by the transitional distribution. On the other hand, with $\Delta\beta_j = 0.0204$, Fig. 9(b) depicts the case whereby majority of the samples have very small weight values (near 0). In fact, from the histogram, approximately 841 out of 1000 samples (i.e. 84.1% of samples) have such small values of associated weights. This illustrates degeneracy.

Based on the description above, it can be seen that the TMCMC sampler is able to generate N samples simultaneously per iteration, whereas the MCMC technique could only compute one new proposal sample per iteration. This implies that the computation performed by the TMCMC algorithm can be easily parallelised whereas MCMC can only be parallelised at chain level. In addition, the TMCMC sampling technique also ensures that the samples in the j^{th} transition step are approximately distributed as per P^j, making the need for burn-in unnecessary [55,136].

The TMCMC sampler algorithm used to generate N samples from a one-dimensional posterior is summarised as follows [55]:

1. Set $j = 0$ and $\beta_j = 0$. Sample $\theta_i \sim P(\theta)$, for $i = 1, \ldots N$.
2. Set $j = j + 1$.
3. Compute β_j using Eq. (17).
4. While $\beta_j < 1$, compute P^j using Eq. (15).
5. Compute $\hat{w}(\theta_i)$ using Eq. (18).
6. Resample $\theta_i \sim \hat{w}(\theta_i)$ for $i = 1, \ldots, N$.
7. Generate N single-step Markov Chains. For each i^{th} chain, generate candidate sample $\theta_i^* \sim q(\theta_i^*|\theta_i)$.
8. Compute the acceptance probability α using Eq. (14) and accept θ_i^* with probability α (see Algorithm 1).
9. Repeat Steps (2) to (8) until $\beta_j = 1$.

To sample from a multi-variate posterior, a component-wise approach is adopted in this paper for the TMCMC sampler. The sampling procedure is the same as that for the mono-dimensional case except that the statistical weight function in Eq. (18) is now defined as:

$$\hat{w}(\boldsymbol{\theta}_i) = \frac{P(D|\boldsymbol{\theta}_i)^{\Delta\beta_j}}{\sum_{i=1}^{N} P(D|\boldsymbol{\theta}_i)^{\Delta\beta_j}}$$

The algorithmic description of the TMCMC sampler in sampling from a general N_d-dimensional posterior is presented in Algorithm 2.

Algorithm 2 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: $\boldsymbol{\theta}_i \sim P(\boldsymbol{\theta})$
  end for
  while $\beta_j < 1$ **do** ▷ Main sampling loop
    Set $j = j + 1$
    Compute $\Delta\beta_j$ using Eq. (17)
    Compute P^j using Eq. (15)
    for $i = 1 : N$ **do** ▷ For each i^{th} chain
      Resample: $\boldsymbol{\theta}_i \sim \hat{w}(\boldsymbol{\theta}_i)$
      for $d = 1 : N_d$ **do** ▷ Update component θ^d
        Draw candidate sample: $\theta_i^{d*} \sim q\left(\theta_i^*|\theta_i^d\right)$
        Accept/Reject θ_i^{d*} using Algorithm 1 with 1 iteration
      end for
    end for
  end while
end procedure

The TMCMC sampler has already been applied in different fields. For instance, the TMCMC technique is employed: to characterize the statistical uncertainties of the spatial variability parameters which are based upon the Cone Penetration Test [138]; to study the multi-modality feature of the Bouc-Wen-Baber-Noori model of hysteresis [139]; to perform model updat-

ing and analyse the uncertainty associated with the creep behavior of soft soil [140]; in performing reliability-based optimization in linear structure designs subjected to random excitations [141]; and in analysing the geometrical uncertainty of a metal frame [142]; in a probabilistic hierarchical Bayesian framework for time-domain model updating [143]; in Bayesian inference for identification of local structural properties of layered composites [46]; to perform cracks identification on beams through Bayesian approach [47]; to perform model parameter updating for piezoelectric energy harvesters [65]; and to perform inverse uncertainty quantification with limited experimental data [42].

3.3. Sequential Monte Carlo Sampler

The Sequential Monte Carlo method (or Particle Filter) [144–146] is a generalized form of the Kalman Filter methods [147,148]. It is mainly employed in system identification [149–152] and utilizes point mass (or "particle") representations of probability densities. The SMC sampler addresses Bayesian inference problems involving a dynamical posterior with continuous observations (e.g. online learning) at every iteration j [153,154]. Over the years, numerous advanced SMC sampling strategies have been proposed such as the block sampling strategies [155], adaptive resampling strategies [156], adaptive SMC sampler [157], and nested SMC strategies [158,159]. However, in this paper, we shall only discuss the basic SMC sampler algorithm proposed by [153] which will be adopted to sample from static posteriors as per the case of the numerical examples presented in this paper.

To provide an understanding of the SMC sampler, this section will first provide an explanation to the procedure behind sampling from a mono-dimensional posterior. At iteration $j = 0$, initial samples of θ_i (for $i = 1, \ldots, N$) are generated from the prior via standard Monte Carlo method following the reasons provided in Section 3.2. Each sample θ_i is assigned an initial statistical (or importance) weight via a statistical weight function $\hat{w}(\theta_i)$ which can be determined using Eq. (18) and setting $\Delta\beta_j = 1$. In Monte Carlo simulation, the method of obtaining samples from a relatively simpler distribution (i.e. the prior in this case) instead of the complex or unknown distribution directly and then assigning weights to the sample in accordance to Eq. (18) is known as Importance sampling [160–162]. Then, for subsequent iterations $j \geqslant 1$, a metric known as the effective sample size, N_{eff}, is calculated by taking the reciprocal of the sum of squares of the statistical weights associated with the samples obtained from previous iteration as illustrated in Eq. (22) [163]:

$$N_{eff} = \frac{1}{\sum_i^N (\hat{w}(\theta_i))^2}$$

N_{eff} provides an indication of degeneracy. If $N_{eff} > \frac{N}{2}$ [153,164], it indicates the absence of degeneracy and the algorithm proceeds directly to the updating step. In the updating step, N independent single-step Markov chains are generated, each initiating from the individual current sample θ_i. For each chain, a candidate sample θ_i^* is sampled from a proposal distribution $q(\theta_i^*|\theta_i)$ which is usually set as some standard distribution such as Normal or multinominal [153,165]. In this paper, $q(\theta_i^*|\theta_i)$ is set as a Normal distribution with mean θ_i and variance σ_p^2 [50]. The variance σ_p^2 also serves as the user-defined tuning parameter of the SMC sampler which influences the spread of candidate samples generated. Each candidate sample θ_i^* becomes the sample generated in the current iteration and its nominal weight is calculated:

$$w(\theta_i^*) = \hat{w}(\theta_i) \cdot \frac{P(\theta_i^*|D)}{P(\theta_i|D)} \cdot \frac{q(\theta_i|\theta_i^*)}{q(\theta_i^*|\theta_i)}$$

Here, $q(\theta_i|\theta_i^*)$ denotes the Backward Markov Kernel which describes the backward transition probability from θ_i^* to θ_i, while $q(\theta_i^*|\theta_i)$ denotes the Forward Markov Kernel which describes the forward transition probability from θ_i to θ_i^*. Both $q(\theta_i|\theta_i^*)$ and $q(\theta_i^*|\theta_i)$ are equivalent as a result of the symmetrical property of the proposal (Normal) distribution [148]. As such Eq. (23) becomes simply:

$$w(\theta_i^*) = \hat{w}(\theta_i) \cdot \frac{P(\theta_i^*|D)}{P(\theta_i|D)}$$

and $\hat{w}(\theta_i)$ is then obtained by normalising Eq. (24):

$$\hat{w}(\theta_i^*) = \frac{w(\theta_i^*)}{\sum_{i=1}^N w(\theta_i^*)}$$

This entire sampling process is repeated for successive iterations j all the way until the termination criteria is reached whereby the desired COV of the estimates of θ is obtained and when the distribution of θ_i becomes stable.

On the other hand, if $N_{eff} < \frac{N}{2}$, it indicates degeneracy. This case is known as resampling, N samples are drawn (with replacement) from the sample set θ_i according to their respective weights $\hat{w}(\theta_i)$. This means that samples with higher weights associated with it gets drawn more frequently, a similar process to weighted random sampling. An illustration to this is provided in Fig. 8. Following which, the statistical weights of these resampled N samples are reset to $\hat{w}(\theta_i) = \frac{1}{N}$. When this is done, the algorithm proceeds to the updating step as explained earlier.

It should be noted that the resampling step does not completely avoid the issue of degeneracy. Instead, it reduces the computation time by eventually discarding samples with insignificant weights associated with it. Moreover, resampling also 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 samples or particles [166], and does not contribute in the exploration of the sample space.

There are 2 notable similarities between TMCMC and the SMC sampler techniques: Both the TMCMC and SMC sampling algorithms involve the assignment of statistical weights to the samples so as to perform weighted random sampling to obtain initial sample points of the single-step MCMC process and update the samples according to the target distribution; and both sampling techniques are able to generate all N samples within an iteration. There are, however, 4 main differences between the two techniques [55,136,153,166]: TMCMC involves the use of transitional distributions in obtaining samples whereas the SMC sampler does not; the SMC sampler addresses degeneracy by performing resampling if $N_{eff} < \frac{N}{2}$ while TMCMC sampler does so by moderating its transition step $\Delta\beta_j$ at every iteration via Eq. (17); for the TMCMC algorithm, candidate samples are either accepted or rejected based on the accept-reject criterion of the MH algorithm whereas for the SMC sampler, there is no rejection of candidate samples; and for the SMC sampler, parallel computing across multiple computer cores is not feasible [154] while this feature is possible for the TMCMC sampler.

The SMC sampler algorithm used to generate N samples from a one-dimensional posterior is summarised as follows [50,164,166]:

1. At iteration $j = 0$, sample $\theta_i \sim P(\theta)$ for $i = 1, \ldots, N$.
2. Calculate $\hat{w}(\theta_i)$ using Eq. (18), setting $\Delta\beta_j = 1$.
3. Set $j = j + 1$. While "termination criteria" = false, calculate N_{eff} using Eq. (22). If $N_{eff} < \frac{N}{2}$, proceed to Step (4). Else, proceed directly to Step (5).
4. Resample $\theta_i \sim \hat{w}(\theta_i)$ for $i = 1, \ldots, N$. Set $\hat{\omega}(\theta_i) = \frac{1}{N}$.
5. Generate N single-step Markov Chains. For each i^{th} chain, generate candidate sample $\theta_i^* \sim q(\theta_i^*|\theta_i)$.
6. Update $w(\theta_i^*)$ using Eq. (24) and obtain $\hat{w}(\theta_i^*)$ using Eq. (25).
7. Repeat steps (2) to (7) until termination criteria is met (i.e. required COV of estimate is obtained and stability of the distribution achieved).

To sample from a multi-variate posterior, a component-wise approach is adopted in this paper for the SMC sampler. The sampling procedure is the same as that for the mono-dimensional case except that the initial statistical weight function is now defined as per Eq. (21) and setting $\Delta\beta_j = 1$. In addition, a multi-variate proposal distribution is used in the form of $q(\boldsymbol{\theta}_i^*|\boldsymbol{\theta}_i) = \prod_{d=1}^{N_d} q\left(\theta_i^{d*}|\theta_i^d\right)$. For this paper, $q(\boldsymbol{\theta}_i^*|\boldsymbol{\theta}_i)$ is set to be a multi-variate Normal distribution with covariance matrix $\boldsymbol{\Sigma}_p$ which takes the form of a square diagonal matrix and now serves as the tuning parameter of the sampler. Furthermore, the updated nominal weight from Eq. (24) is now defined as:

$$w(\boldsymbol{\theta}_i^*) = \hat{w}(\boldsymbol{\theta}_i) \cdot \frac{P(\boldsymbol{\theta}_i^*|D)}{P(\boldsymbol{\theta}_i|D)}$$

whereby $\boldsymbol{\theta}_i^* = \left\{\theta_i^{1*}, \theta_i^{2*}, \ldots, \theta_i^{N_d-1*}, \theta_i^{N_d*}\right\}$ is the updated vector states of all the i^{th} samples across d dimensions, $P(\boldsymbol{\theta}_i^*|D)$ is the posterior value evaluated at $\boldsymbol{\theta}_i^*$, and $P(\boldsymbol{\theta}_i|D)$ is the posterior value evaluated at $\boldsymbol{\theta}_i$. From there, $\hat{w}(\boldsymbol{\theta}_i^*)$ is obtained by normalising $w(\boldsymbol{\theta}_i^*)$ as per Eq. (25). To explain the sampling procedure from a general N_d-dimensional posterior, we define the general updated d^{th} state vector $\boldsymbol{\theta}_i^{d*} = \left\{\theta_i^{1*}, \ldots, \theta_i^{d-1*}, \theta_i^{d*}, \theta_i^{d+1}, \ldots, \theta_i^{N_d}\right\}$. Using this convention, the algorithmic description of the SMC sampler in sampling from a general N_d-dimensional posterior is presented in Algorithm 3.

Algorithm 3 Component-wise SMC 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: $\boldsymbol{\theta}_i \sim P(\boldsymbol{\theta})$
    Compute $\hat{w}(\boldsymbol{\theta}_i)$ using Eq. (21)
  end for
  while "Termination criteria" = false **do** ▷ Main sampling loop
    Set $j = j + 1$
    Compute N_{eff} using Eq. (22)
    if $N_{eff} < \frac{N}{2}$ **then** ▷ Conditional resampling step
      for $i = 1 : N$ **do**

Resample set: $\boldsymbol{\theta}_i \sim \hat{w}(\boldsymbol{\theta}_i)$
Reset $\hat{w}(\boldsymbol{\theta}_i) = \frac{1}{N}$
end for
end if
for $i = 1 : N$ **do** ▷ For each i^{th} chain
for $d = 1 : N_d$ **do** ▷ Update component θ^l
Sample: $\theta_i^{d*} \sim q\left(\theta_i^{d*}|\theta_i\right)$
Update state vector $\boldsymbol{\theta}_i^{d*}$
end for
Set $\boldsymbol{\theta}_i^* = \left\{\theta_i^{1*}, \theta_i^{2*}, \ldots, \theta_i^{N_d-1*}, \theta_i^{N_d*}\right\}$
Update $w(\boldsymbol{\theta}_i^*)$ using Eq. (26)
Obtain $\hat{w}(\boldsymbol{\theta}_i^*)$ using Eq. (25)
end for
end while
end procedure

Currently, the SMC sampler has already been employed to: perform multi-resolution alignment of multiple unsynchronised audio sequences [167]; analyse and quantify the uncertainty of the measured data from probabilistic nonlinear state-space models of dynamical systems [168]; perform cosmological parameter estimation using Approximate Bayesian Computation in large high dimensional and correlated parameter spaces [53]; and to estimate the parameter of interest in hydrological models whereby non-linear dependency structures as well as multiple nodes are often present [169]; analyse and quantify the uncertainty of the measured data from probabilistic nonlinear state-space models of dynamical systems [168]; to perform uncertainty reduction in prognostics [170]; to estimate parameters of dynamical systems from big data [50]; to perform Bayesian learning of state-space models with highly informative observations [171]; and to extract bearing fault features via Bayesian approach [172].

4. Applications

4.1. Spring-mass system under static force: Linear relationship between measured parameter and uncertain parameter

Fig. 10 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:

$$F = -k \times d$$

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

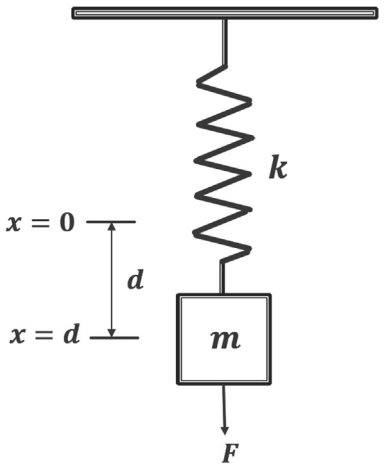


Fig. 10. Schematic diagram of the simple spring-mass system.

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

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

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

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 Eqs. (28) and (29) as well as in reality. However, we will assume that the contribution of measurement "noise" comes only from $F_{measured}$. 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 scatter plot shown in Fig. 11 while its numerical values are presented in a table as shown in Table 3.

4.1.1. Linear least-squares method

One direct way to solve for k analytically would be via the method of Linear Least-squares minimization [173]. The equations to the Linear Least-squares method is as follows:

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

whereby $\mathbf{x}$ is the design matrix, which in this case would be the vector of the displacement values, $\mathbf{x}^T$ is the transpose of the design matrix, and F is the vector of the real measurement values of the force acting on the spring. As such, k can be solved by re-expressing Eq. (30) into the following form:

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

Using the data values shown in Table 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 Fig. 12.

4.1.2. Bayesian model updating

For this problem, the a priori knowledge of k is based on the initial hypothesis that k can range between 0.01 N/m and 1000 N/m. As such, this non-informative prior distribution of $k, P(k)$, can be modelled after a Uniform distribution whose lower-bound and upper-bound values are 0.01 N/m and 1000 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:

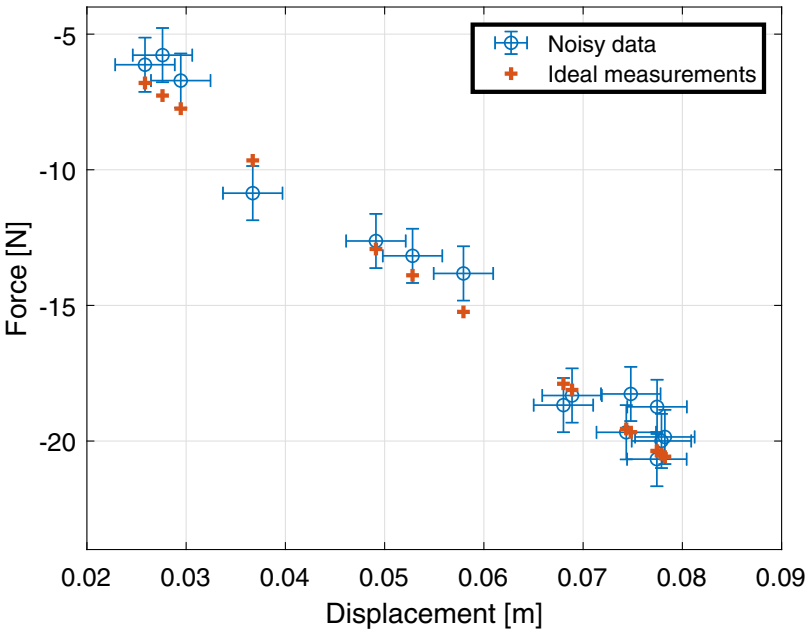


Fig. 11. Scatter plot of the 15 simulated "noisy" data of Force against the respective values of Displacements.

Table 3
Numerical values of the data illustrated in Fig. 11.

Displacement (m)	Noisy Data (N)	Ideal Measurement (N)
0.0259	−6.13	−6.80
0.0276	−5.77	−7.26
0.0295	−6.71	−7.75
0.0367	−10.86	−9.65
0.0491	−12.63	−12.92
0.0528	−13.17	−13.89
0.0579	−13.82	−15.24
0.0680	−18.68	−17.89
0.0688	−18.32	−18.12
0.0743	−19.68	−19.55
0.0748	−18.26	−19.67
0.0774	−20.67	−20.36
0.0775	−18.74	−20.37
0.0779	−20.00	−20.49
0.0782	−19.85	−20.58

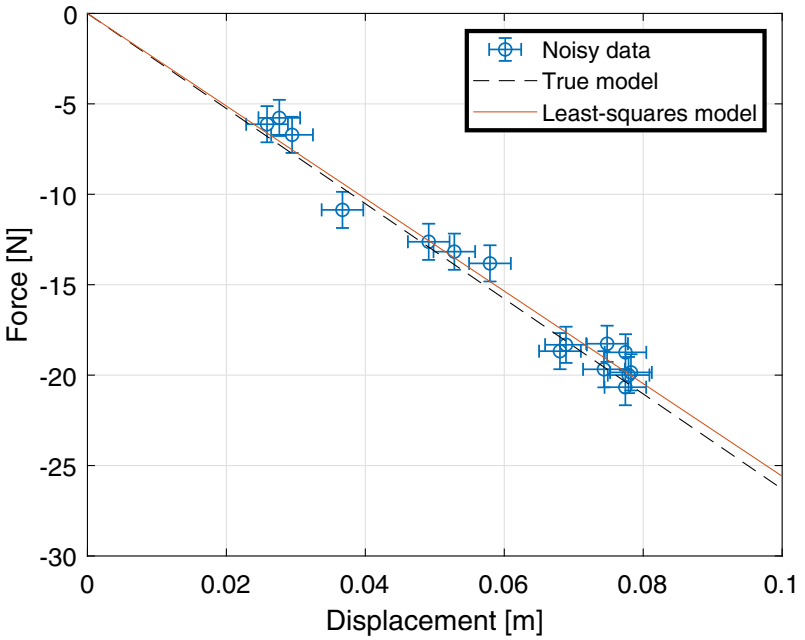


Fig. 12. The updated linear model via the linearization method as illustrated by the red line with the "noisy" data represented by the blue circles. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

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

The justification behind the choice of the likelihood function and its associated standard deviation is as presented in Section 1.

In this problem, we will compare the performances of the MCMC, TMCMC, and the SMC sampler in attaining a COV of less than 2.0% in their respective estimation of k. This comparison will be done on the basis of the sample size required to attain convergence, the time-elapsed, and the COV in the estimation of k.

4.1.3. 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 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 4 while the corresponding time-series sample plots are presented in Fig. 31. From the time-series sample plots in Fig. 31, the burn-in length $N_{burn-in}$ is obtained and presented in Table 4. Based on the results, the optimum value of tuning parameter is determined to be 22.50 N/m.

The the main sampling procedure is then conducted with a sample size of 1000. 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 693.44 N/m and sampling is first done with 0 burn-in. The resulting time-series plot and histogram of the sample values of k is shown in Fig. 13. Based on the time-series plot in Fig. 13, the burn-in length is determined to be approximately 75. Fig. 14 illustrates the resulting time-series plot and histogram of the sample values of k after discarding the first 75 samples. The sampler took 0.23 seconds of computation time and yielded an estimated mean value of 255.64 N/m for k with a standard deviation σ of 4.23 N/m, thus, giving the estimate a COV of 1.65%.

The above procedures were conducted using only a single MCMC chain. Further simulation experiments are performed using multiple chains to observe if there are any significant changes in the precision of the estimated values of k. For this purpose, a set of simulations is performed using 3 and 5 chains, each chain having a sample size of 1000 and burn-in length of 75 as determined from Fig. 13. The tuning parameter value is set at 22.50 N/m. The numerical results are summarized in Table 5. As a form of comparison, 2 more sets of simulations were performed using a single MCMC chain with sample size 3000 and 5000 with burn-in length of 100 and 20 respectively. The numerical results are presented in Table 6.

From the results in Tables 5 and 6, it can be seen that by varying the number of MCMC chains, the COV of the estimate does not show much variation whereas the time elapsed for the simulation is significantly shorter when using multiple chains with smaller sample size compared to using a single chain with a larger sample size. A reason to account for this is due to the fact that each individual MCMC chain is computed independently and in parallel by the sampler.

4.1.4. TMCMC sampler results

For the TMCMC sampler, a sample size of 1000 samples was generated from the posterior to ensure sufficient convergence of the sample estimate of k. The burn-in length is set to be 0. The sampler took 11.85 seconds of computation time over 5 iterations and yielded an estimated mean value of 256.01 N/m for k with a standard deviation of 4.26 N/m, thus, giving the estimate a COV of 1.66%. The resulting time-series sample plot and histogram are presented in Fig. 15.

Table 4
Summary of results from varying the tuning parameter values while keeping the sample size fixed at 10000.

Tuning Parameter (N/m)	Acceptance Level	Time (sec)	$N_{burn-in}$
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

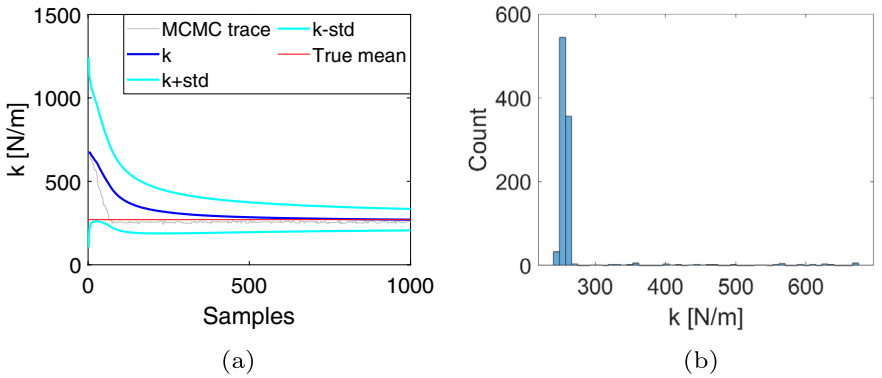


Fig. 13. The resulting time-series sample plot (a) and the histogram (b) obtained using MCMC sampler with sample size 1000 with 0 burn-in. The red line in the time-series sample plot denotes the true sample mean value. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

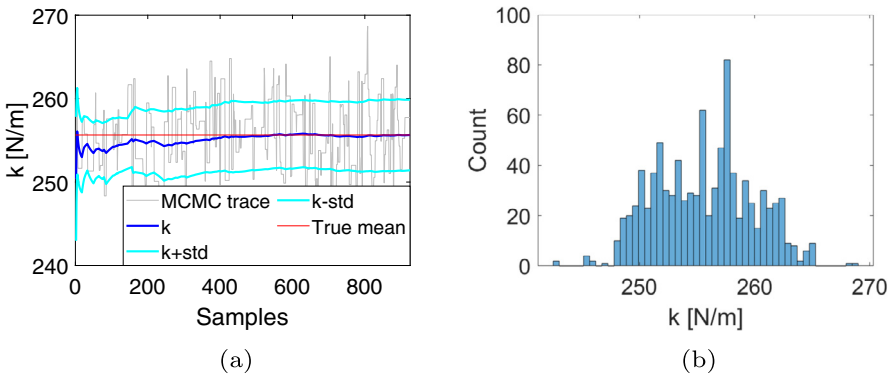


Fig. 14. The resulting time-series sample plot (a) and the histogram (b) after discarding the first $N_{burn-in} = 75$ samples. The red line in the time-series sample plot denotes the true sample mean value. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5
Summary of results from varying the number of chains, each chain with sample size of 1000 and burn-in length of $N_{burn-in} = 75$. Reference solution: $k = 263$ N/m.

N_{chain}	$N_{burn-in}$	$E[k]$ (N/m)	$\sigma[k]$ (N/m)	COV (%)	Time (sec)	Iterations
1	75	255.64	4.23	1.65	0.23	1000
3	75	255.55	4.20	1.64	0.38	1000
5	75	255.48	4.19	1.64	0.43	1000

Table 6
Summary of results from varying the number of samples. Reference solution: $k = 263$ N/m.

$N_{samples}$	$N_{burn-in}$	$E[k]$ (N/m)	$\sigma[k]$ (N/m)	COV (%)	Time (sec)	Iterations
3000	100	255.96	4.22	1.65	0.58	3000
5000	20	255.71	4.25	1.66	0.96	5000

4.1.5. SMC sampler results

Before the main sampling procedure is conducted, a calibration experiment is performed to determine the value of the tuning parameter such that the SMC sampler is able to achieve a CoV of less than 2.0% in its estimation of k. To do this, a fixed sample size of 10000 is generated from the posterior using 5 different values of the tuning parameter: $0.01\ N^2/m^2$, $1.50\ N^2/m^2$, $7.50\ N^2/m^2$, $15.00\ N^2/m^2$, and $30.00\ N^2/m^2$. The number of iteration of the SMC sampler is set at 1 given that the sampling is done from a static posterior. The numerical results are summarized in Table 7.

It is noteworthy that the choice of tuning parameter has barely any impact on the computation time by the SMC sampler. However, based on the results presented in Table 7, it can been observed that the larger the value of the tuning parameter, the larger the COV associated with the estimate of k. This is due to the fact that for a larger tuning parameter value, the spread of the candidate samples obtained from the proposal distribution becomes larger. To achieve the aforementioned criteria on the COV of the estimate, the tuning parameter value to be used for this problem will be 1.50.

The main sampling procedure is then conducted with a sample size of 1000. This value of sample size is chosen as it ensures sufficient convergence of the sample estimate of k by the SMC sampler. The sampler took 0.26 seconds of computation time and yielded an estimated mean value of 255.13 N/m for k with a standard deviation 4.45 N/m, thus, giving the estimate a COV of 1.74%. The resulting time-series sample plot and histogram are presented in Fig. 16.

4.1.6. Discussions

The overall results of the sampling estimates of k for each sampler are summarized in Table 8 and the resulting Bayesian model update by each of the sampler are also presented in Fig. 17. Based on the results, it can be seen that for the same number of samples obtained from the posterior, the computation time elapsed for the TMCMC sampler is significantly higher compared to that of the MCMC and SMC sampler. This is due to the fact that the TMCMC algorithm is such that for each iter-

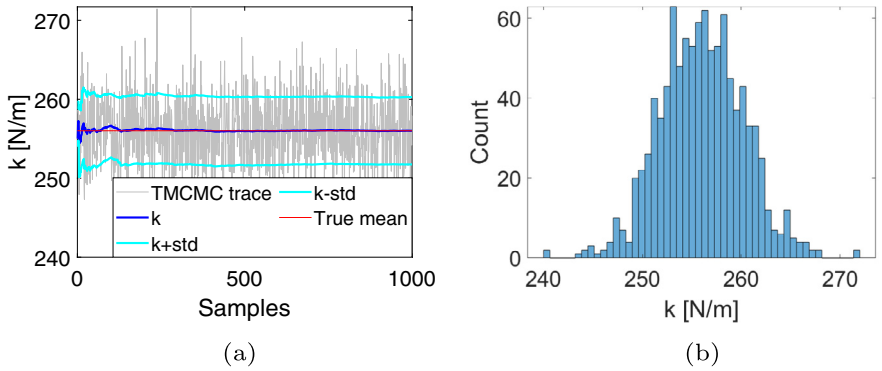


Fig. 15. The resulting time-series sample plot (a) and the histogram (b) obtained using TMCMC sampler with sample size 1000. The red line in the time-series sample plot denotes the true sample mean value. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 7
Summary of results of SMC sampler from varying the tuning parameter values while keeping the sample size fixed at 10000. Reference solution: $k = 263\ N/m$

Tuning Parameter (N^2/m^2)	$E[k]$ (N/m)	$\sigma[k]$ (N/m)	CoV (%)	Time (sec)
0.01	255.65	4.15	1.62	2.14
1.50	255.81	4.22	1.65	2.14
7.50	255.50	4.83	1.89	2.14
15.00	256.31	5.67	2.21	2.15
30.00	255.94	7.05	2.76	2.14

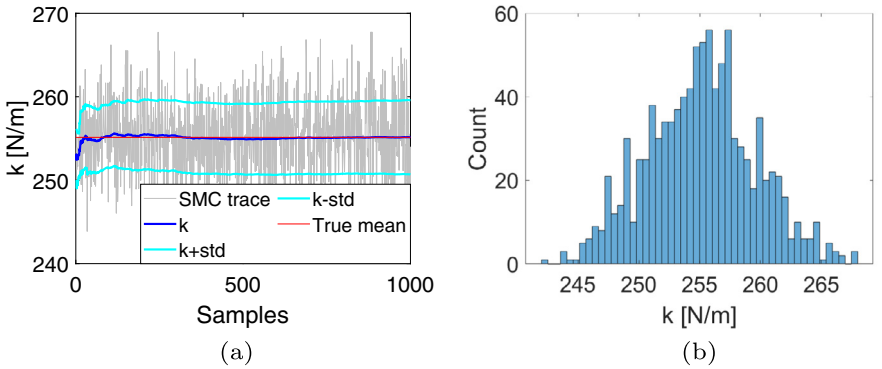


Fig. 16. The resulting time-series sample plot (a) and the histogram (b) obtained using SMC sampler with sample size 1000. The red line in the time-series sample plot denotes the true sample mean value. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

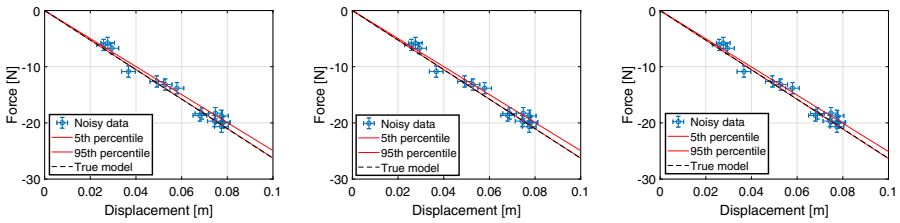
ation, it obtains $N_{samples}$ samples from the transitional distribution, where $N_{samples}$ is the sample size to be obtained from the posterior, as mentioned in Section 3.2. In addition, it can also be seen from the results that the estimate of k obtained by MCMC sampler yields the least COV and has the shortest computation time which would make the MCMC sampler the better choice of sampler among the 3 samplers for this problem.

4.2. Inverse eigenvalue problem: 2-D bi-modal posterior

In this example, the performance of the advanced Monte Carlo 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, H, which takes on the following form:

Table 8
Summary of the numerical results of the estimation of k by the respective samplers. Reference solution: $k = 263\ N/m$

Method	$N_{samples}$	$E[k]$ (N/m)	$\sigma[k]$ (N/m)	COV (%)	Time (sec)	Iterations
MCMC	925	255.64	4.23	1.65	0.23	1000
TMCMC	1000	256.01	4.26	1.66	11.85	5
SMC	1000	255.13	4.45	1.74	0.26	1



(a) MCMC model update. (b) TMCMC model update (c) SMC model update

Fig. 17. Results of the model updating for the respective samplers. The red lines denote the 5^{th} and 95^{th} percentile bounds. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

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

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 [174]. In this problem, θ_1 and θ_2 are the matrix elements which are assumed to have fixed values, $\hat{\theta_1} = 0.5$ and $\hat{\theta_2} = 1.5$.

In a physical context, the square matrix H represents a physical quantity whose eigenvalues λ_1 and λ_2 represent the possible observations that can be made for that physical quantity. 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:

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

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

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, we will simulate 15 independent "noisy" data from each model to perform the analysis. The available data are presented in the form of a scatter plot shown in Fig. 18 while its numerical values are presented in a table as shown in Table 9.

4.2.1. 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, this non-informative prior distribution, $P(\theta_1, \theta_2)$, 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:

$$P(\boldsymbol{\lambda}|\boldsymbol{\theta}) \propto exp\left[-\frac{1}{2}\sum_{i=1}^{2}\sum_{n=1}^{15}\left(\frac{\lambda_{i,n}^{noisy} - \lambda_i^{model}}{\sigma_i}\right)^2\right]$$

whereby $\boldsymbol{\lambda}$ is the 15 by 2 vector of the "noisy" observations, $\boldsymbol{\theta}$ is the vector of the uncertain model parameters (θ_1 and θ_2), and σ_i is the standard deviation of ϵ_i for $i = 1, 2$.

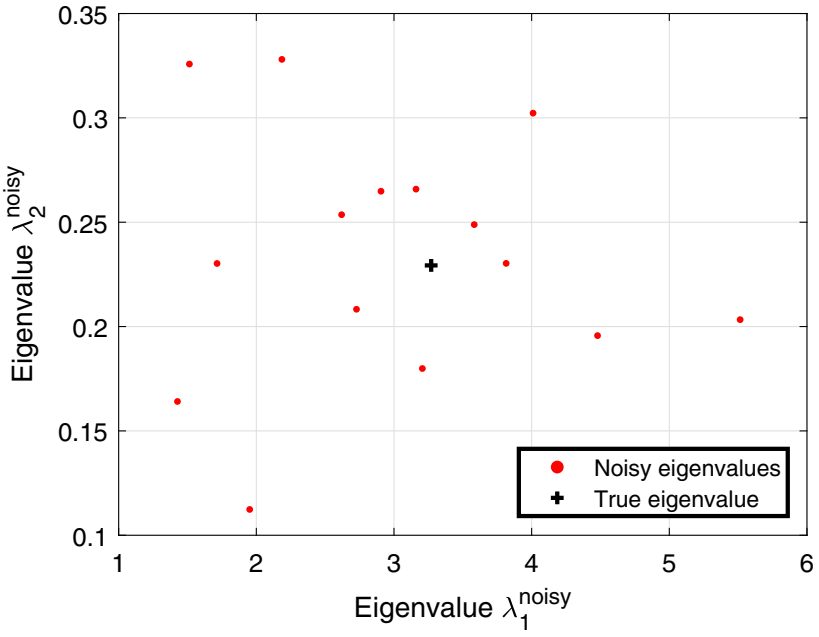


Fig. 18. Scatter plot of the 15 different measured values of λ_1^{noisy} and λ_2^{noisy}.

Table 9
Numerical values of the "noisy" data illustrated in Fig. 18.

Noisy eigenvalue 1 (λ_1^{noisy})	Noisy eigenvalue 2 (λ_2^{noisy})
1.51	0.33
4.01	0.30
3.16	0.27
3.21	0.18
2.19	0.33
1.71	0.23
2.73	0.21
5.51	0.20
1.95	0.11
4.48	0.20
1.43	0.16
2.91	0.26
3.81	0.23
3.58	0.25
2.62	0.25

4.2.2. MCMC sampler results

The main sampling procedure is performed with a sample size of 1000 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 0.235. 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 scatter plot are presented in Fig. 19. To ensure sufficient burn-in, the burn-in length is set to be 30 and the resulting scatterplot matrix and 2D scatter plot as shown in Fig. 20 where it can be observed that the samples converge about $\{\theta_1, \theta_2\}$ = {0.51, 1.36}. The sampler took 0.40 seconds of computation time over 1000 iterations.

4.2.3. TMCMC sampler results

For the TMCMC sampler, a sample size of 1000 samples was obtained from the posterior distribution to ensure sufficient convergence of the sample estimate of θ_1 and θ_2. The burn-in length is set to be 0. The sampler took 18.38 seconds of com-

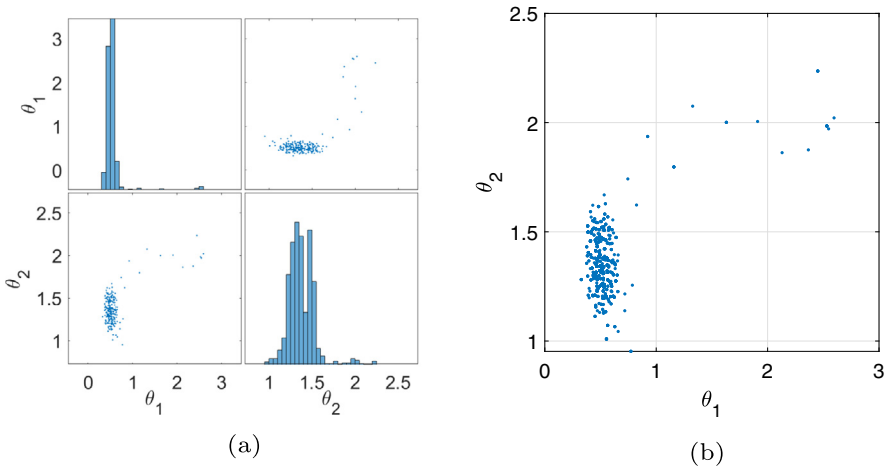


Fig. 19. The resulting scatterplot matrix (a) and 2*D* scatter plot (b) obtained using MCMC sampling with sample size 1000 and 0 burn-in.

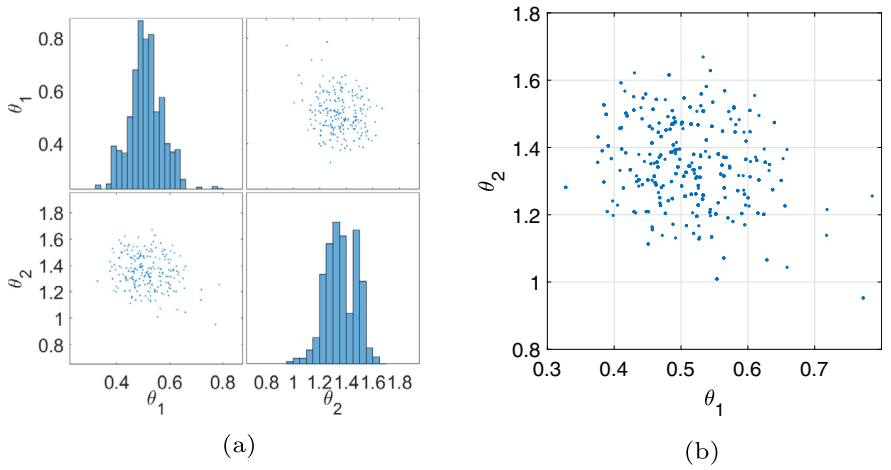


Fig. 20. The resulting scatter plot matrix (a) and 2*D* scatter plot (b) after discarding the first $N_{burn-in} = 30$ samples.

putation time over 5 iterations and the resulting scatterplot matrix and 2*D* scatter plot are presented in Fig. 21. Based on Fig. 21, the scatter plot features two distinct convergence points centered about $\{\theta_1, \theta_2\} = \{0.51, 1.35\}$ and $\{2.75, 0.27\}$.

4.2.4. SMC sampler results

Before the main sampling procedure is conducted, a calibration experiment is performed by varying the tuning parameter to see its effect on the scatter plots. To do this, a fixed sample size of 10000 is used for sampling using 4 different covariance matrix: $0.001 \cdot I, 0.1 \cdot I, I$, and $10 \cdot I$. The number of iteration of the SMC sampler is set at 1 given that the sampling is done from a static posterior distribution. The results are illustrated in the form of a series of scatter plot matrix diagrams for each covariance matrix used as tuning parameter in Fig. 22.

Based on the Figure, it can be seen that as the tuning parameter is being scaled by a larger factor, spread of the samples become wider. In addition, it can also be observed that when the Identity matrix is used as the tuning parameter, the SMC sampler is no longer able to resolve the two peaks of the posterior and any tuning parameter that is scaled larger than I itself would also give rise to the same result. In such cases, more iterations are required before the samples converge to the posterior. To ensure that the samples converge to the posterior and identify the two peaks within an iteration, $0.001 \cdot I$ is used as the choice of the tuning parameter for this problem.

The main sampling procedure is then conducted with a sample size of 1000. The sampler took 0.63 seconds of computation time over 1 iteration and the resulting scatterplot matrix and 2*D* scatter plot are presented in Fig. 23. Based on Fig. 23, the scatter plot features two distinct convergence points centered about $\{\theta_1, \theta_2\} = \{0.52, 1.35\}$ and $\{2.86, 0.26\}$.

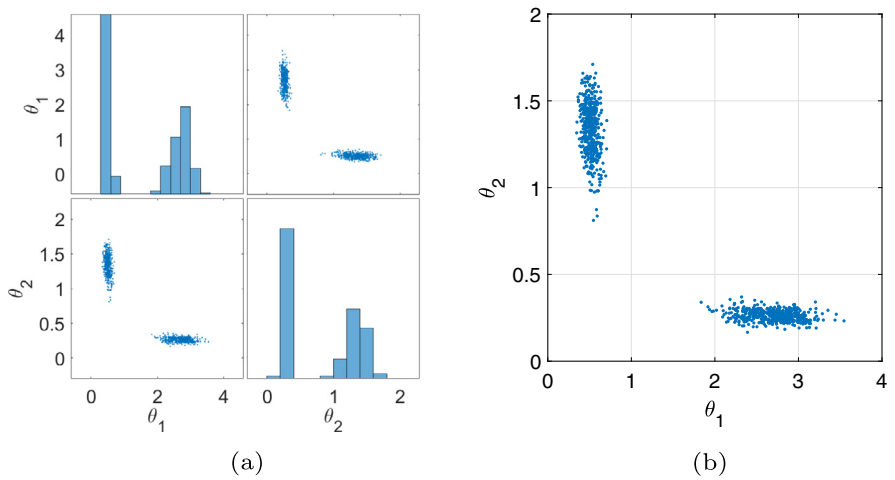
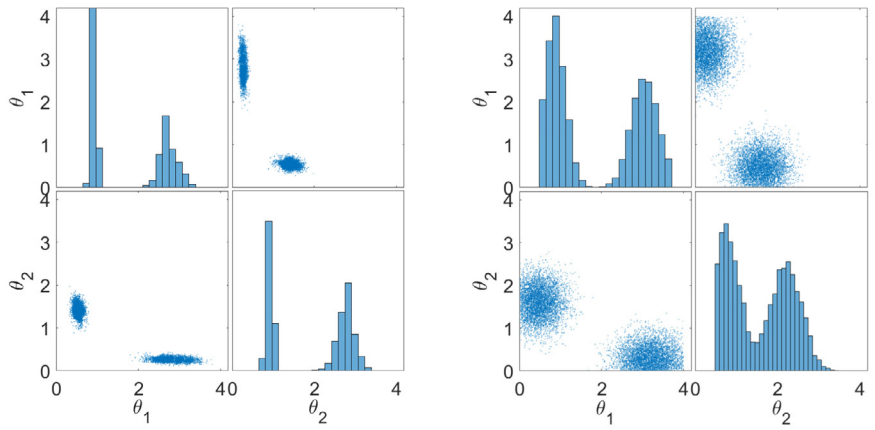
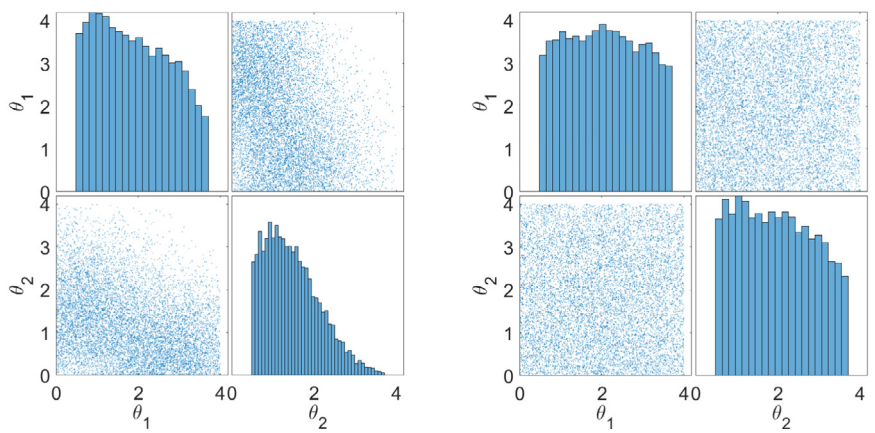


Fig. 21. The resulting 2*D* scatter plot matrix (a) and scatter plot (b) obtained using TMCMC sampler with sample size 1000.



(a) Tuning parameter $= 0.001 \cdot I$ (b) Tuning parameter $= 0.1 \cdot I$



(c) Tuning parameter $= I$ (d) Tuning parameter $= 10 \cdot I$

Fig. 22. The resulting scatter plot matrix diagrams for different covariance matrix used as the tuning parameter.

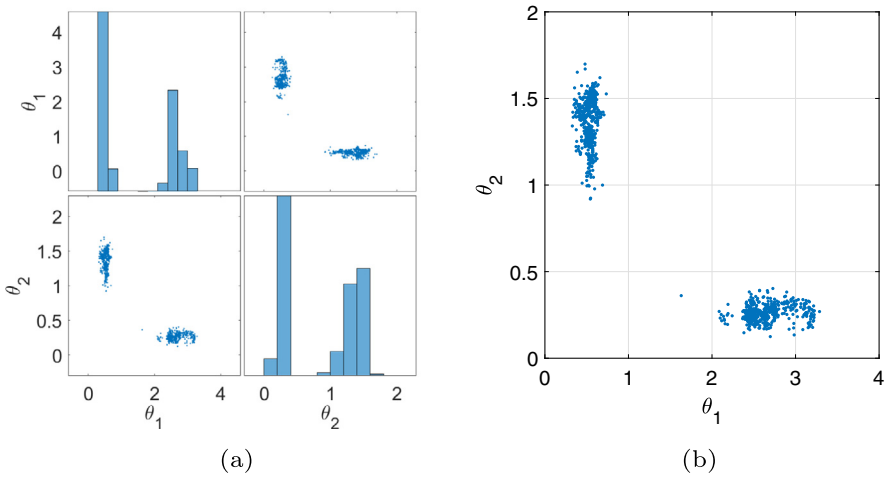


Fig. 23. The resulting 2*D* scatter plot matrix (a) and scatter plot (b) obtained using SMC sampler with sample size 1000.

4.2.5. 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 [175] when discussing the implementation of the algorithm on a posteriors with multiple peaks. The TMCMC and SMC 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 sampler, this is attributed to the resampling procedure whereby samples closer to each of the two peaks of the posterior are resampled with higher probability according to Eq. (21). 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 MCMC, TMCMC, and SMC samplers are presented in Fig. 24.

4.3. Bayesian model updating of DLR-AIRMOD test structure

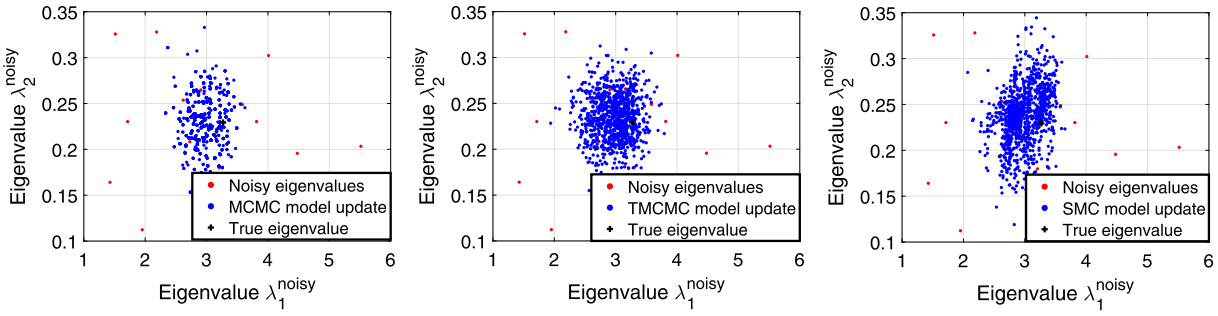
In order to investigate the existing variability of the natural response frequency in the dynamic behaviour of nominally identical test structures the Göttingen's German Aerospace Centre (DLR) constructed a replica of the GARTEUR SM-AG19 benchmark known as AIRcraft MODel (AIRMOD) [176]. The DLR-AIRMOD is an aluminum structure consisting of aluminium beams connected using bolted joints. Details of the AIRMOD structure and the experimental settings can be found in [177–179].

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 two locations for variability. 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 10.

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 11.

In [179], a deterministic model updating via the sensitivity method [180,181] was performed using the information provided in Table 11 to update the 18 input uncertain model parameters listed in Table 10. The resulting statistics of the 18 updated parameters are summarised in Table 12.

In this section, the Bayesian model updating approach is be adopted to update the 18 input parameters which will be done using MCMC, TMCMC and SMC 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 complex, higher-dimensional posterior. The model to be used for the Bayesian updating procedure is a surrogate



(a) MCMC model update. (b) TMCMC model update (c) SMC model update

Fig. 24. Updated scatter plot profiles obtained from: (a) MCMC, (b) TMCMC, and (c) SMC sampling methods.

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. The ANN model consists of 14 individual ANNs, each trained to predict a specific frequency output of the corresponding active mode [48]. These 14 individual ANN each 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 [48]. In this example, the simulation and computation was implemented using the OpenCossan software [182,183].

4.3.1. Bayesian model updating

Contrary to the standard procedure of using prior identified from previous experience or methods (see e.g. [179]), here the non-informative prior is used. Therefore, for each of the individual input parameter, θ_i (for $i = 1, \ldots, 18$), $P(\theta_i)$, 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, $P(\boldsymbol{\theta})$, can be expressed as follows:

$$P(\boldsymbol{\theta}) = \prod_{i=1}^{18} P(\theta_i)$$

The likelihood function is a 14-dimensional multivariate Normal distribution. Assuming independence between the experimental outputs, it is mathematically expressed as follows [179]:

Table 10
List of the 18 input parameters and their respective details. Data obtained from [177].

Parameter	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 11
Test statistics of the 14 frequency outputs to be used to perform model updating. Data obtained from [177].

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

$$P(f|\boldsymbol{\theta}) \propto \prod_{i=1}^{14} exp\left[-\frac{(f_i - M_i)^2}{2 \cdot \sigma_i^2}\right]$$

whereby f_i is the experimental measurement of the i^{th} frequency mode, M_i is the i^{th} ANN model used to predict the frequency output of the i^{th} active mode, and σ_i is the standard deviation of the residual between the experimental result f_i and the model $\widehat{M}_i$. Here, σ_i 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.

4.3.2. MCMC sampler results

A nominal covariance matrix, *COV*, 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 12. 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 $10^{-3} \cdot COV$. A sample size of 1500 is obtained from the posterior, with a burn-in length of 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 13

Table 12
Updated statistics of the 18 input parameters obtained using the Sensitivity model updating method. Results taken from [179].

Parameter	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

while the posterior distribution of the normalised data for each input parameter is presented in a scatter plot matrix as illustrated in Fig. 25.

4.3.3. TMCMC sampler results

Using the TMCMC sampler, a sample size of 1000 samples was obtained from the posterior. The burn-in length for all iterations, up to the second last iteration, was set at 50 while the burn-in length for the last iteration was set at 200. The simulation was performed using local parallelization across 34 cores, each with a CPU memory of 10.5 Gigabytes. The computation involved a total of 22 iterations over 5 h 13 min. The resulting statistics of the updated input parameters are summarised in Table 13 while the posterior of the normalised data for each input parameter are presented in a scatter plot matrix as illustrated in Fig. 26.

4.3.4. SMC sampler results

For the SMC sampler, the tuning parameter used is $10^{-3} \cdot COV$ and the sampling procedure was executed to obtain 1000 samples from the posterior. The simulation was performed on 1 core with a CPU memory of 10.5 Gigabytes. The computation involved a total of 1 iteration over 84.35 seconds. The resulting statistics of the updated input parameters are summarised in Table 13 and while the posterior distribution of the normalised data for each input parameter are presented in a scatter plot matrix as illustrated in Fig. 27. The experiment was also conducted using 20 iterations which yielded similar results to that shown in Fig. 27.

4.3.5. Discussions

Table 13 summarises the numerical results of the estimates obtained by the respective samplers. As a form of evaluating the model updating performance between the three samplers, the scatter plot profiles illustrating the distribution of the experimental samples will be compared against those from the updated ANN surrogate model. For simplicity, the comparison between frequencies f_1 and f_4, and between frequencies f_6 and f_7 are in Figs. 28–30. Figs. 32–34 in Section A.2 show the scatter plot profile comparison for all frequencies.

To quantify the closeness and similarity level between the scatter plot profiles of the experimental frequency samples and the sample output obtained from the updated model, the two-sample Kolmogorov–Smirnov (KS) test [184–186] is used as the metric. The two-sample KS test tests the null hypothesis that two given samples come from the same continuous distribution against the alternative hypothesis that they do not. The two-sided test is performed at 5% significance level for each of the 14 frequency outputs. Table 14 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 14 shows that the two-sample KS test indicates that there is no similarity between the distributions of the experimental frequency samples and the sample output from the updated model for all 14 active frequencies tested. This is supported from Fig. 28 where it can be seen that frequency scatter plots profile from the updated model (in blue) obtained using MCMC sampling technique do not show any similarity to that of the experimen-

Table 13
Updated statistics of the 18 input parameters obtained using MCMC, TMCMC, and SMC samplers.

Parameter	MCMC Mean	COV (%)	TMCMC Mean	COV (%)	SMC Mean	COV (%)
θ_1	1.13×10^3	1.08	1.89×10^3	14.59	1.86×10^3	55.74
θ_2	8.98×10^3	0.29	7.64×10^3	8.97	7.65×10^3	55.12
θ_3	1.27×10^2	8.69	1.85×10^2	5.19	1.32×10^2	55.83
θ_4	5.70×10^1	0.76	5.25×10^1	44.76	7.20×10^1	55.07
θ_5	3.52×10^1	1.23	3.59×10^1	54.57	7.21×10^1	55.00
θ_6	1.20×10^7	0.45	1.47×10^6	29.98	1.05×10^7	53.79
θ_7	1.32×10^9	2.08	1.03×10^9	52.63	1.02×10^9	54.19
θ_8	2.49×10^{-1}	0.28	1.92×10^{-1}	19.54	2.05×10^{-1}	54.95
θ_9	4.07×10^{-2}	0.76	1.99×10^{-1}	6.26	1.91×10^{-1}	54.12
θ_{10}	2.48×10^{-1}	0.20	1.95×10^{-1}	5.26	1.92×10^{-1}	54.60
θ_{11}	1.13×10^{-2}	7.82	2.82×10^{-2}	5.21	1.53×10^{-2}	55.66
θ_{12}	1.21×10^{-2}	2.45	1.03×10^{-2}	38.76	1.60×10^{-2}	52.87
θ_{13}	2.81×10^{-2}	0.05	1.17×10^{-2}	63.88	1.53×10^{-2}	55.14
θ_{14}	1.89×10^7	0.74	3.61×10^7	8.17	2.04×10^7	55.47
θ_{15}	2.72×10^7	0.37	1.04×10^7	15.03	2.07×10^7	54.79
θ_{16}	6.99×10^6	0.21	2.36×10^6	23.08	7.14×10^6	54.86
θ_{17}	1.00×10^7	3.45	3.49×10^7	66.60	4.96×10^7	57.23
θ_{18}	4.24×10^7	0.15	5.01×10^7	55.69	5.24×10^7	54.63

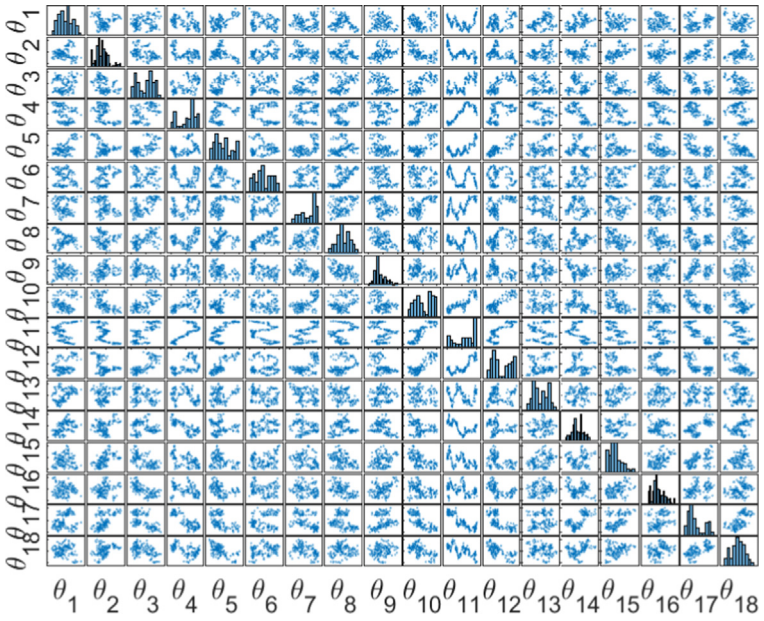


Fig. 25. Scatter plot matrix of the 18 inputs obtained using MCMC sampler with 1500 samples and burn-in length of 500. The data presented here are normalised to take values between 0 and 1.

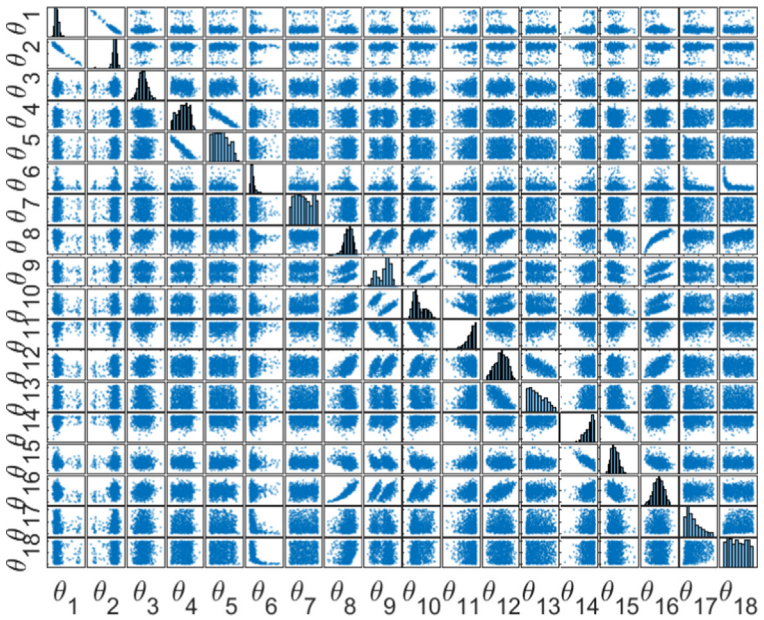


Fig. 26. Scatter plot matrix of the 18 inputs obtained using TMCMC sampler with 1000 samples and burn-in length of 50 for the first 21 iterations and 200 for the last iteration. The data presented here are normalised to take values between 0 and 1.

tal frequencies (in red). This is especially so in Fig. 28(b) where it is evident that the frequency scatter plots profile of the updated model deviates significantly from those of the experimental frequencies. 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 distribution, especially when the posterior is only concentrated within a small area of the entire sample space. This comes despite the COV of the estimation of the 18 input parameters all fall below 10% as seen in Table 13. 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

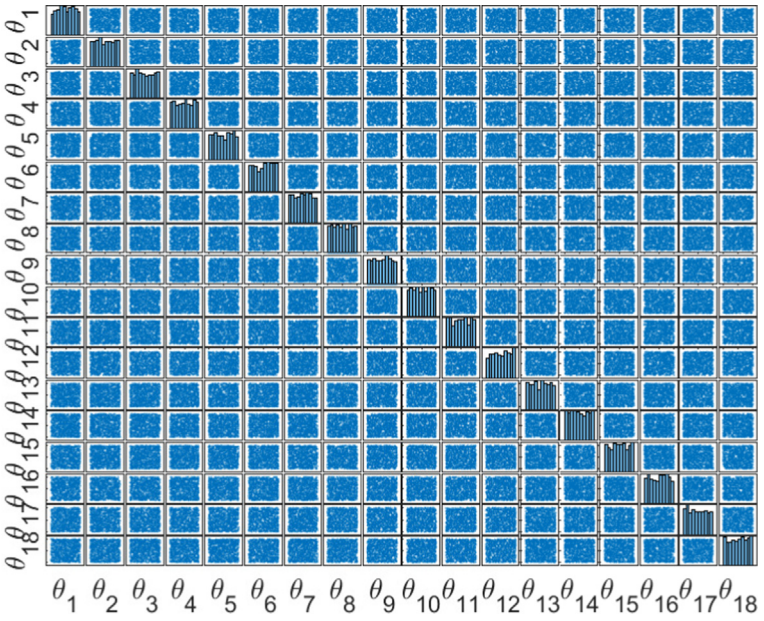


Fig. 27. Scatter plot matrix of the 18 inputs after Bayesian model updating using 1000 samples obtained via SMC sampling method. The data presented here are normalised to take values between 0 and 1.

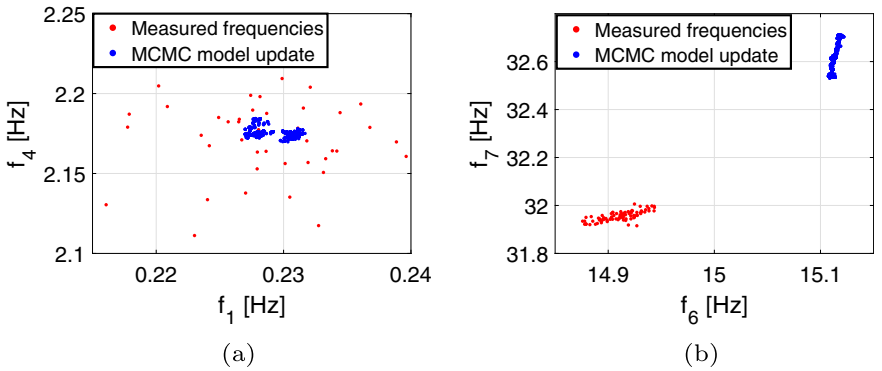


Fig. 28. Comparison between the scatter plot obtained from experimental frequency data and that of from the updated ANN surrogate model via MCMC sampler. Subplot (a) presents the comparison between frequencies f_1 and f_4, while subplot (b) presents the comparison between frequencies f_6 and f_7.

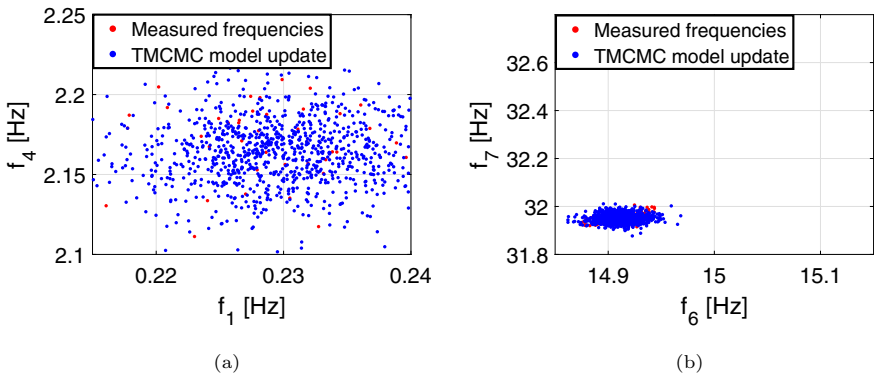
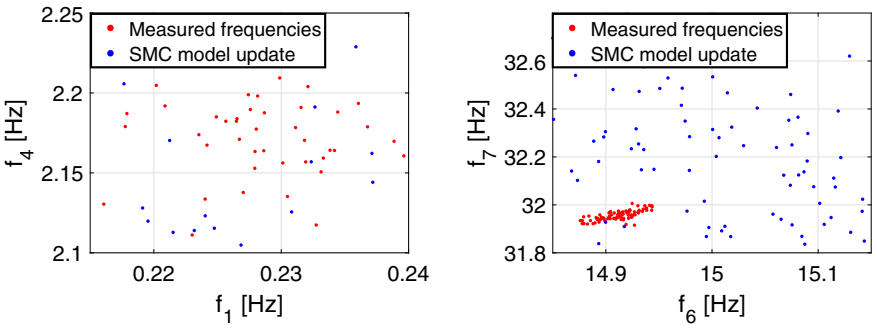


Fig. 29. Comparison between the scatter plot obtained from experimental frequency data and that of from the updated ANN surrogate model via TMCMC sampler. Subplot (a) presents the comparison between frequencies f_1 and f_4, while subplot (b) presents the comparison between frequencies f_6 and f_7.



(a) (b)

Fig. 30. Comparison between the scatter plot obtained from experimental frequency data and that of from the updated ANN surrogate model via SMC sampler. Subplot (a) presents the comparison between frequencies f_1 and f_4, while subplot (b) presents the comparison between frequencies f_6 and f_7.

Table 14
P-values and test indicator from the two-sample KS test performed on the frequency samples from the updated model by MCMC, TMCMC, and SMC samplers along with the frequency samples obtained from experiment.

Output	Mode name	MCMC p-value	Indicator	TMCMC p-value	Indicator	SMC p-value	Indicator
f_1	RBM Yaw	6.73×10^{-5}	1	2.67×10^{-1}	0	5.47×10^{-20}	1
f_2	RBM Roll	2.19×10^{-67}	1	1.10×10^{-1}	0	5.80×10^{-34}	1
f_3	RBM Pitch	5.03×10^{-69}	1	1.52×10^{-1}	0	4.87×10^{-25}	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	2.70×10^{-54}	1
f_6	3nWingBending	2.86×10^{-71}	1	1.75×10^{-1}	0	1.63×10^{-32}	1
f_7	WingTorsionAnti	2.86×10^{-71}	1	6.84×10^{-2}	0	1.96×10^{-17}	1
f_8	WingTorsionSym	2.86×10^{-71}	1	6.50×10^{-3}	1	1.72×10^{-51}	1
f_9	4nWingBending	2.86×10^{-71}	1	5.13×10^{-1}	0	9.32×10^{-43}	1
f_{10}	1nWingForeAft	2.86×10^{-71}	1	5.24×10^{-8}	1	2.07×10^{-16}	1
f_{11}	2nWingForeAft	2.86×10^{-71}	1	3.06×10^{-58}	1	5.41×10^{-46}	1
f_{12}	VtpTorsion	2.86×10^{-71}	1	4.77×10^{-4}	1	2.73×10^{-63}	1
f_{13}	2nHtpBending	2.86×10^{-71}	1	2.16×10^{-4}	1	1.89×10^{-16}	1
f_{14}	HtpForeAft	2.86×10^{-71}	1	1.53×10^{-5}	1	2.70×10^{-54}	1

results obtained using Sensitivity model updating [179] (see Table 12) 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.

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 distribution 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 Fig. 29 that the frequency scatter plot profile for the updated model mostly coincides with the frequency scatter plot profile for the experimental frequencies. Coupled with the fact that the algorithm is free from any tuning parameters, this makes the TMCMC sampler the most suitable choice of sampler among the three algorithms for such problem.

For the case of the SMC sampler, Table 14 shows that the two-sample KS test indicates that there is no similarity between the distributions of the frequency samples and the sample output from the updated model for all 14 active frequencies tested. This is observed from Fig. 30 where it can be seen that while the frequency scatter plots from the updated model generally encompass the frequency scatter plots from the experimental data, there is no similarity in the scatter plot profiles between the two entities. In addition, it can also be observed from Fig. 30(b) that the scatterplot profile from the updated model shows a larger spread compared to that of the experimental data. This is attributed to the estimates of the 18 updated input parameters having high COVs (all above 50%) as shown in Table 13 and that their respective posteriors illustrate a small degree of update from the Uniform prior as seen in Fig. 27. This is due to the ineffectiveness in the importance sampling technique (see Eq. (18)) when applied in high-dimensional cases [55,187–189]. In addition, like the MCMC sampler, the

efficiency of the SMC sampler in attaining convergence of samples within the least iterations is also dependent on the choice of the user-defined tuning parameter. The choice of tuning parameter used was made based on the results from the sensitivity model updating. This makes the SMC sampler inefficient when such prior information is unavailable.

It has to be noted that the intention of this experiment was not to obtain the optimum updating results as such work was previously done and presented in the literature by Govers *et. al* (2015) [177]. Rather, the purpose of this section is to highlight the difference in the model updating performance between MCMC, TMCMC, and SMC samplers for a high-dimensional system.

5. Further discussions

The three sampling techniques reviewed in this paper have been applied to 4 case studies with different inherent challenges. In summary, the MH algorithm is relatively the easiest to implement among the three 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 [55] 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 4.2. 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 [55,179] of the entire sample space as shown in the case example presented in Section 4.3. These short-comings, however, are addressed with recent developments of the algorithm such as the Adaptive Metropolis–Hastings (AMH) algorithm [135], Adaptive Metropolis-within-Gibbs (AMWG) algorithm [190,191], Lam *et. al*'s proposed multi-level MCMC approach [192] which seeks to improve the algorithm's exploration of the sample space by dividing the sampling process into multiple levels, as well as the TMCMC algorithm [55].

The TMCMC sampler algorithm is a tune-free algorithm in that it does not require any tuning parameter which makes it a relatively convenient choice of sampler especially for "Black-box" problems. Its key strength lies in the ability of the algorithm to effectively sample from multi-modal posteriors as seen in Section 4.2 as well as high-dimensional posterior as seen in Section 4.3 which makes the TMCMC a robust sampler [55]. 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. burn-in however, may need to be considered when sampling from relatively complex higher-dimension posteriors, especially in cases when the posterior takes up only a small area of the entire sample space. 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 three 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 sampler algorithm, like the TMCMC sampler algorithm, lessens the consideration for a burn-in period due to the initialization procedure of the algorithm which obtains an initial set of samples directly from the prior distribution thereby ensuring that the final samples obtained are within the posterior. The SMC sampler is also able to sample from a multi-modal posterior shown in Section 4.2 where it is also able to identify the two peaks of a bi-modal posterior distribution. Furthermore, in sampling from static posterior, the sampling time elapsed by the SMC sampler is generally comparable to that of the MCMC algorithm despite its relative complexity in its algorithm and this is due to the algorithm obtaining samples directly from the posterior rather than transitional distributions as per the case of TMCMC. The efficiency of the SMC sampler, however, lies in the choice of the value of the tuning parameter. As seen in Sections 4.1, and 4.2, we have demonstrated that the larger the value and scale of the tuning parameter chosen, the less precise the initial estimate by the SMC sampler thereby requiring more than one iteration to attain the required convergence. In addition to this, as observed in Section 4.2, we have also demonstrated that the larger the scale of the covariance matrix as the tuning parameter, the less the SMC sampler is able to resolve the distinct peaks of the bi-modal posterior within an iteration. And furthermore, the SMC sampler is less efficient and effective in sampling from a high-dimensional posterior [188,189] as shown in Section 4.3 which is attributed to the inefficiency and inapplicability of the Importance sampling procedure to samples in high dimensions [187]. To overcome this issue, one can turn to advanced SMC sampling strategies such as the through the use of an adaptive MCMC mutation kernel proposed in [193], or the nested SMC sampling approach [158,159].

It is also observed from the case examples that different sampling techniques yield different statistics of the posterior distribution (i.e. the posterior mean and its variance). 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 final posterior distribution [97]. This assumption however, falls short when dealing with multi-modal posteriors as seen in Section 4.2. The TMCMC sampler assumes that the samples would eventually converge to the final posterior distribution by

sampling from a series of intermediate transitional distributions [55]. 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 4.2 and 4.3. Finally, the SMC sampler assumes the statistics of the posterior can be approximated through the combination Importance sampling and Resampling procedure. This assumption is valid in instances whereby the posterior distribution has a low number of dimensions (e.g. less than 5 [14]) and becomes inapplicable for high-dimensional cases (i.e. 18 dimensions) as seen in Section 4.3 when Importance sampling fails [187]. 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.

6. 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 this tutorial paper, we have presented the concept behind three advanced sampling techniques: Markov Chain Monte Carlo, Transition Markov Chain Monte Carlo, and Sequential Monte Carlo sampling. The presented algorithms have been applied to solve three 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 5. 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 this paper 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 [71,147,50,166,194,195]). 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 this paper. Some approaches dealing with this type of problems for finance applications can be found in references [196,197]. In addition to this, the recorded data themselves may not necessarily be independently identically distributed, as it was assumed in this paper. Furthermore, to broaden the generality of problems that could be encountered, the model relating the measured variables D and the uncertain model parameter(s) $\boldsymbol{\theta}$ may not even be known precisely (i.e. model uncertainty; see [198,56,57,199]). These are currently active research areas.

For the benefit of the readers, the presented algorithms and the examples discussed in this paper are freely available as part of the OpenCossan software [182,183] on GitHub:https://github.com/cossan-working-group/BayesianModelUpdating

CRediT authorship contribution statement

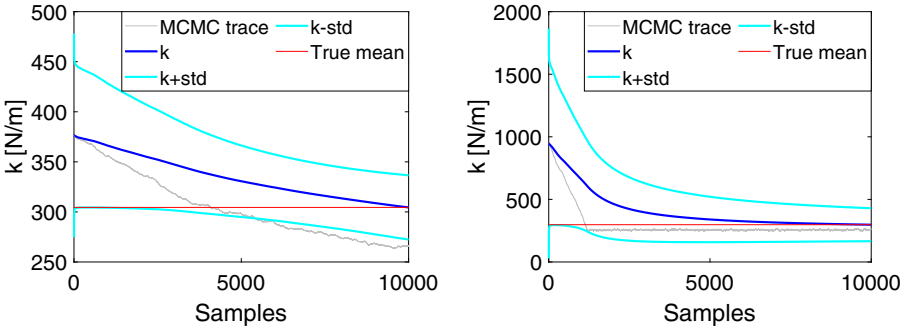
Adolphus Lye: Methodology, Conceptualization, Investigation, Software, Writing - original draft. **Alice Cicirello:** Supervision, Conceptualization, Writing - review & editing. **Edoardo Patelli:** Supervision, Conceptualization, Software, Writing - review & editing, Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

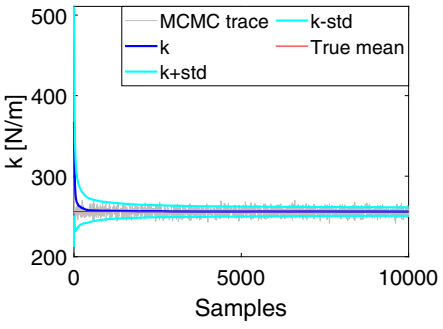
Appendix A

A.1. Additional results for the 1-D spring-mass system case study

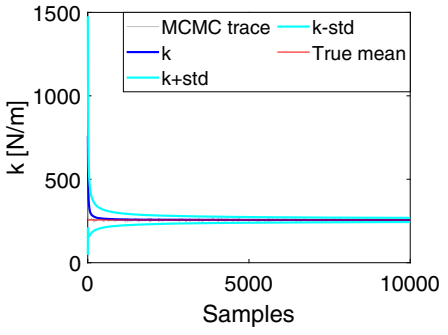


(a) Tuning parameter = 0.01 N/m

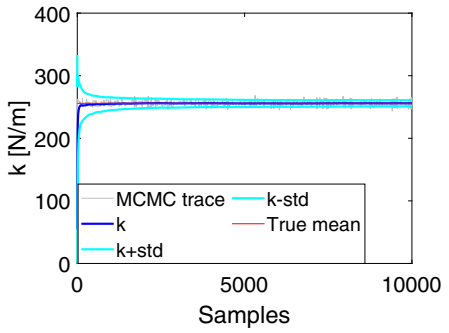
(b) Tuning parameter = 1.50 N/m



(c) Tuning parameter = 22.50 N/m



(d) Tuning parameter = 80.00 N/m



(e) Tuning parameter = 100.00 N/m

Fig. 31. The resulting time-series sample plots for different values of tuning parameter for the MCMC sampler obtained using a sample size of 10000 with 0 Burn-in length.

A.2. Additional results for the 18-D DLR-AIRMOD problem

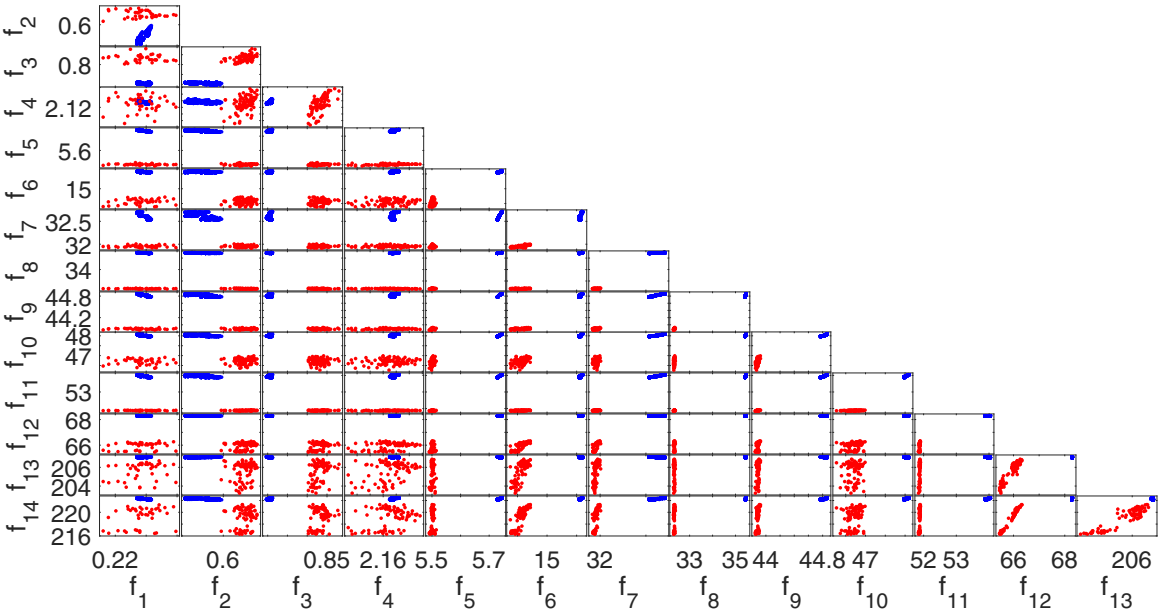


Fig. 32. The scatterplot matrix illustrating the updated model output profile obtained using MCMC. The blue scatter plots represent the frequency output from the updated model while the red scatter plots represent the experimental frequency measurements.

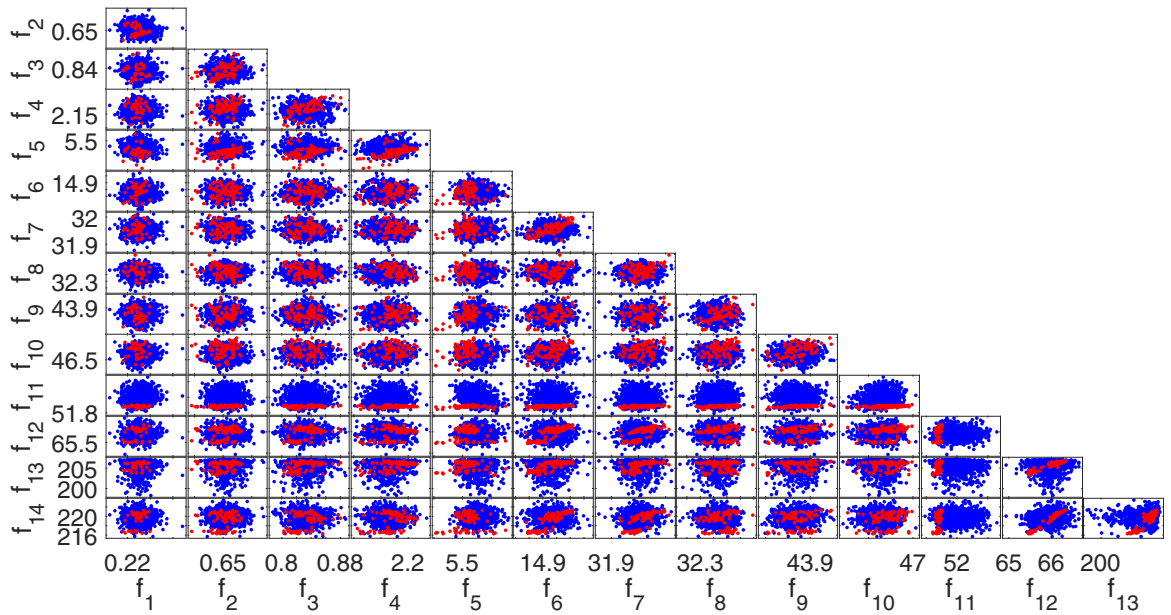


Fig. 33. The scatterplot matrix illustrating the updated model output profile obtained using TMCMC. The blue scatter plots represent the frequency output from the updated model while the red scatter plots represent the experimental frequency measurements. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

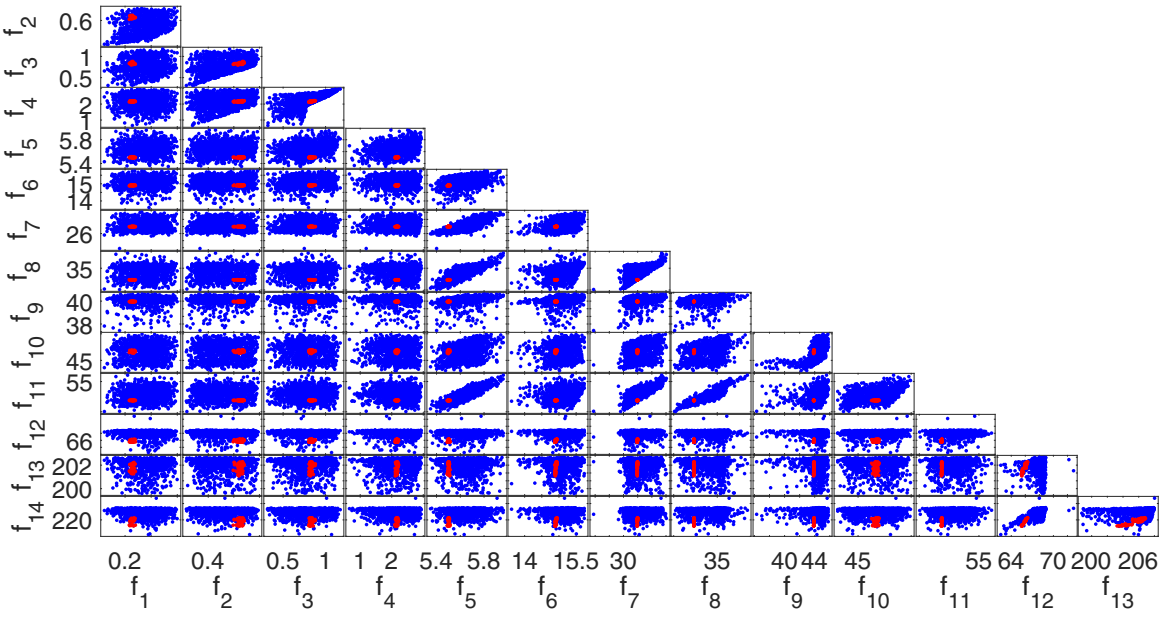


Fig. 34. The scatterplot matrix illustrating the updated model output profile obtained using SMC. The blue scatter plots represent the frequency output from the updated model while the red scatter plots represent the experimental frequency measurements.

References

[1] K. Worden, Nonlinearity in Structural Dynamics: Detection, Identification and Modelling, CRC Press, 2019.
[2] M.I. Friswell, J.E. Mottershead, Finite Element Model Updating in Structural Dynamics, Springer, Dordrecht, 2011, ISBN 978-9048145355.
[3] H. Waisman, E. Chatzi, 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 82 (2009) 303–328, https://doi.org/10.1002/nme.2766.
[4] I. Hlaváček, J. Chleboun, I. Babuška, Reality, Mathematics, and Computation, North-Holland Series in Applied Mathematics and Mechanics Uncertain Input Data Problems and the Worst Scenario Method (2004) 1–49, https://doi.org/10.1016/s0167-5931(04)80005-6.
[5] J. Helton, An exploration of alternative approaches to the representation of uncertainty in model predictions, Reliability Engineering and System Safety 85 (2004) 39–71, https://doi.org/10.1016/s0951-8320(04)00051-1.
[6] J. Helton, Uncertainty and sensitivity analysis in the presence of stochastic and subjective uncertainty, Journal of Statistical Computation and Simulation 57 (1997) 3–76, https://doi.org/10.1080/00949659708811803.
[7] R.L. Winkler, Uncertainty in probabilistic risk assessment, Reliability Engineering and System Safety 54 (1996) 127–132, https://doi.org/10.1016/s0951-8320(96)00070-1.
[8] Z. Meng, Y. Pang, Y. Pu, 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 363 (2020), https://doi.org/10.1016/j.cma.2020.112886 112886.
[9] Y. Luo, A. Li, Z. Kang, Reliability-based design optimization of adhesive bonded steel–concrete composite beams with probabilistic and non-probabilistic uncertainties, Engineering Structures 33 (2011) 2110–2119, https://doi.org/10.1016/j.engstruct.2011.02.040.
[10] B. Ni, C. Jiang, X. Han, An improved multidimensional parallelepiped non-probabilistic model for structural uncertainty analysis, Applied Mathematical Modelling 40 (2016) 4727–4745, https://doi.org/10.1016/j.apm.2015.11.047.
[11] B. Ni, C. Jiang, Z. Huang, Discussions on non-probabilistic convex modelling for uncertain problems, Applied Mathematical Modelling 59 (2018) 54–85, https://doi.org/10.1016/j.apm.2018.01.026.
[12] M. Rashki, Hybrid control variates-based simulation method for structural reliability analysis of some problems with low failure probability, Applied Mathematical Modelling 60 (2018) 220–234, https://doi.org/10.1016/j.apm.2018.03.009.
[13] H.H. Khodaparast, J.E. Mottershead, K.J. Badcock, Interval model updating with irreducible uncertainty using the Kriging predictor, Mechanical Systems and Signal Processing 25 (2011) 1204–1226, https://doi.org/10.1016/j.ymssp.2010.10.009.
[14] J.L. Beck, L.S. Katafygiotis, Updating models and their uncertainties. I: Bayesian statistical framework, Journal of Engineering Mechanics 124 (1998) 455–461, https://doi.org/10.1061/(asce)0733-9399(1998)124:4(455).
[15] L.S. Katafygiotis, J.L. Beck, Updating models and their uncertainties. II: Model identifiability, Journal of Engineering Mechanics 124 (1998) 463–467, https://doi.org/10.1061/(asce)0733-9399(1998)124:4(463).
[16] L.S. Katafygiotis, C. Papadimitriou, H. Lam, A probabilistic approach to structural model updating, Soil Dynamics and Earthquake Engineering 17 (1998) 495–507, https://doi.org/10.1016/S0267-7261(98)00008-6.
[17] H.F. Lam, L.S. Katafygiotis, 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 130 (2004) 34–48, https://doi.org/10.1061/(asce)0733-9399(2004)130:1(34).
[18] S. Au, Connecting Bayesian and frequentist quantification of parameter uncertainty in system identification, Mechanical Systems and Signal Processing 29 (2012) 328–342, https://doi.org/10.1016/j.ymssp.2012.01.010.
[19] D. Giagopoulos, A. Arailopoulos, V. Dertimanis, C. Papadimitriou, E. Chatzi, K. Grompanopoulos, Structural health monitoring and fatigue damage estimation using vibration measurements and finite element model updating, Structural Health Monitoring 18 (2019) 1189–1206, https://doi.org/10.1177/1475921718790188.
[20] J. Ching, J.L. Beck, New Bayesian model updating algorithm applied to a structural health monitoring benchmark, Structural Health Monitoring: An International Journal 3 (2004) 313–332, https://doi.org/10.1177/1475921704047499.
[21] K.V. Yuen, J. Beck, L.S. Katafygiotis, Efficient model updating and health monitoring methodology using incomplete modal data without mode matching, Structural Control Health Monitoring 13 (2006) 91–107, https://doi.org/10.1002/stc.144.

[22] S.E. Azam, C. Papadimitriou, 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 (2014)..
[23] J.P. Noel, M. Schoukens, Cross-fertilising research in nonlinear system identification between the mechanical, control and machine learning fields: Editorial statement, Mechanical Systems and Signal Processing 130 (213–220) 2019. doi:10.1016/j.ymssp.2019.04.061..
[24] G. Capellari, E. Chatzi, S. Mariani, Optimal sensor placement through bayesian experimental design: Effect of measurement noise and number of sensors, Proceedings 1 (2016) 41, https://doi.org/10.3390/ecsa-3-d006.
[25] P.G. Daneshmand, R. Jafari, A 3D hybrid BE–FE solution to the forward problem of electrical impedance tomography, Engineering Analysis with Boundary Elements 37 (2013) 757–764, https://doi.org/10.1016/j.enganabound.2013.01.016.
[26] Z. Xu, Q. Li, 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 (2010) 42–50, https://doi.org/10.1007/978-3-642-15615-1_6.
[27] Y. Xu, F. Dong, Galerkin boundary element method for the forward problem of ERT, Flow Measurement and Instrumentation 21 (2010) 172–177, https://doi.org/10.1016/j.flowmeasinst.2009.12.004.
[28] T. Shi, J. Ma, H. Huang, Y. Qiu, H. Zeng, Z. Li, D. Qian, Variance reduction for global response problem based on forward Monte Carlo calculation, Nuclear Engineering and Design 322 (2017) 291–300, https://doi.org/10.1016/j.nucengdes.2017.07.006.
[29] A. Elidrysy, S. Harir, A. Zouhair, Y. Boughaleb, Simulation of an extended 3D mixed Ising model by Monte Carlo method, Materials Today: Proceedings 30 (2020) 993–997, https://doi.org/10.1016/j.matpr.2020.04.370.
[30] S. Li, L. Caracoglia, Surrogate Model Monte Carlo simulation for stochastic flutter analysis of wind turbine blades, Journal of Wind Engineering and Industrial Aerodynamics 188 (2019) 43–60, https://doi.org/10.1016/j.jweia.2019.02.004.
[31] G. Dahlquist, B. Ake, N. Anderson, Numerical Methods, Dover, Mineola (New York), 2003, ISBN 978-0486428079.
[32] J.H. Albert, Teaching Bayesian statistics using sampling methods and MINITAB, The American Statistician 47 (1993) 182–191, https://doi.org/10.2307/2684973.
[33] C.E. Papadopoulos, H. Yeung, Uncertainty estimation and monte carlo simulation method, Flow Measurement and Instrumentation 12 (2001) 291–298, https://doi.org/10.1016/s0955-5986(01)00015-2.
[34] N.T. Thomopoulos, Essentials of Monte Carlo Simulation, Springer, 2015, ISBN 978-1461460213.
[35] E. Simoen, G.D. Roeck, G. Lombaert, Dealing with uncertainty in model updating for damage assessment: A review, Mechanical Systems and Signal Processing 56–57 (2015) 123–149, https://doi.org/10.1016/j.ymssp.2014.11.001.
[36] R.J. Rossi, Mathematical statistics: An introduction to likelihood based inference, Hoboken, John Wiley and Sons, NJ, 2018, ISBN 978-1118771044.
[37] H. Karimi, 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 67 (2014) 178–198, https://doi.org/10.1016/j.compchemeng.2014.04.007.
[38] Z. Szabo, X. Liu, L. Xiang, Semiparametric sieve maximum likelihood estimation for accelerated hazards model with interval-censored data, Journal of Statistical Planning and Inference 205 (2020) 175–192, https://doi.org/10.1016/j.jspi.2019.07.002.
[39] F. Li, K. Li, K. Lu, Z. Li, Random noise suppression and parameter estimation for Magnetic Resonance Sounding signal based on maximum likelihood estimation, Journal of Applied Geophysics 176 (2020), https://doi.org/10.1016/j.jappgeo.2020.104007 104007.
[40] K.V. Yuen, Bayesian Methods for Structural Dynamics and Civil Engineering, John Wiley and Sons Asia, Singapore, 2010, ISBN 978-0470824559.
[41] T. Bayes, 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 53 (1763) 370–418. doi:10.1098/rstl.1763.0053..
[42] M. Faes, M. Broggi, E. Patelli, Y. Govers, J. Mottershead, M. Beer, D. Moens, A multivariate interval approach for inverse uncertainty quantification with limited experimental data, Mechanical Systems and Signal Processing 118 (2019) 534–548, https://doi.org/10.1016/j.ymssp.2018.08.050.
[43] E.T. Jaynes, Information theory and statistical mechanics, Stanford, CA: Microwave Laboratory, W.W. Hansen Laboratories of Physics, Stanford University, 1957.
[44] E.T. Jaynes, The relation of bayesian and maximum entropy methods, Maximum-Entropy and Bayesian Methods in Science and Engineering (1988) 25–29, https://doi.org/10.1007/978-94-009-3049-0_2.
[45] R.D. Teloli, S.D. Silva, T.G. Ritto, G. Chevallier, Bayesian model identification of higher-order frequency response functions for structures assembled by bolted joints, Mechanical Systems and Signal Processing 151 (2021), https://doi.org/10.1016/j.ymssp.2020.107333 107333.
[46] W.J. Yan, D. Chronopoulos, S.C. Chinchilla, K.V. Yuen, 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 143 (2020), https://doi.org/10.1016/j.ymssp.2020.106802 106802.
[47] S. He, C.T. Ng, Guided wave-based identification of multiple cracks in beams using a bayesian approach, Mechanical Systems and Signal Processing 84 (2017) 324–345, https://doi.org/10.1016/j.ymssp.2016.07.013.
[48] E. Patelli, M. Broggi, Y. Govers, J.E. Mottershead, Model updating strategy of the DLR-AIRMOD test structure, Procedia Engineering 199 (2017) 978–983, https://doi.org/10.1016/j.proeng.2017.09.221.
[49] T.G. Ritto, R. Sampaio, R.R. Aguiar, Uncertain boundary condition bayesian identification from experimental data: a case study on a cantilever beam, Mechanical Systems and Signal Processing 68–69 (2016) 176–188, https://doi.org/10.1016/j.ymssp.2015.08.010.
[50] P. Green, S. Maskell, Estimating the parameters of dynamical systems from big data using sequential Monte Carlo samplers, Mechanical Systems and Signal Processing 93 (2018) 379–396, https://doi.org/10.1016/j.ymssp.2016.12.023.
[51] H. Jalali, H.H. Khodaparast, H. Madinei, M. Friswell, Stochastic modelling and updating of a joint contact interface, Mechanical Systems and Signal Processing 129 (2019) 645–658, https://doi.org/10.1016/j.ymssp.2019.04.003.
[52] T. Rogers, K. Worden, 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 140 (2020), https://doi.org/10.1016/j.ymssp.2019.106580 106580.
[53] E. Jennings, M. Madigan, astroABC: An Approximate Bayesian Computation Sequential Monte Carlo sampler for cosmological parameter estimation, Astronomy and Computing 19 (2017) 16–22, https://doi.org/10.1016/j.ascom.2017.01.001.
[54] Y. Wang, Y. Ni, X. Wang, Real-time defect detection of high-speed train wheels by using Bayesian forecasting and dynamic model, Mechanical Systems and Signal Processing 139 (2020), https://doi.org/10.1016/j.ymssp.2020.106654 106654.
[55] J.Y. Ching, Y.C. Chen, Transitional Markov Chain Monte Carlo method for Bayesian model updating, model class selection, and model averaging, Journal of Engineering Mechanics 133 (2007), https://doi.org/10.1061/(ASCE)0733-9399(2007)133:7(816).
[56] H. Rappel, L. Beex, L. Noels, S. Bordas, Identifying elastoplastic parameters with Bayes' theorem considering output error, input error and model uncertainty, Probabilistic Engineering Mechanics 55 (2019) 28–41, https://doi.org/10.1016/j.probengmech.2018.08.004.
[57] B. Goller, G. Schuëller, Investigation of model uncertainties in Bayesian structural model updating, Journal of Sound and Vibration 330 (2011) 6122–6136, https://doi.org/10.1016/j.jsv.2011.07.036.
[58] E. Simoen, C. Papadimitriou, G. Lombaert, On prediction error correlation in Bayesian model updating, Journal of Sound and Vibration 332 (2013) 4136–4152, https://doi.org/10.1016/j.jsv.2013.03.019.
[59] B. Han, T.Y. Xiang, H.B. Xie, A Bayesian inference framework for predicting the long-term deflection of concrete structures caused by creep and shrinkage, Engineering Structures 142 (2017) 46–55, https://doi.org/10.1016/j.engstruct.2017.03.055.
[60] Z.P. Bazant, J.C. Chern, Bayesian statistical prediction of concrete creep and shrinkage, ACI Journal Proceedings 81 (1984) 319–330, https://doi.org/10.14359/10686.
[61] H. Keitel, A. Dimmig-Osburg, L. Vandewalle, L. Schueremans, Selecting creep models using Bayesian methods, Materials and Structures 45 (2012) 1513–1533, https://doi.org/10.1617/s11527-012-9854-x.

[62] X. Guan, R. Jha, Y. Liu, Model selection, updating, and averaging for probabilistic fatigue damage prognosis, Structural Safety 33 (2011) 242–249, https://doi.org/10.1016/j.strusafe.2011.03.006.
[63] Q. Huang, P. Gardoni, S. Hurlebaus, A probabilistic damage detection approach using vibration-based nondestructive testing, Structural Safety 38 (2012) 11–21, https://doi.org/10.1016/j.strusafe.2012.01.004.
[64] W.J. Yan, L.S. Katafygiotis, A novel Bayesian approach for structural model updating utilizing statistical modal information from multiple setups, Structural Safety 52 (2015) 260–271, https://doi.org/10.1016/j.strusafe.2014.06.004.
[65] P. Peralta, R.O. Ruiz, A.A. Taflanidis, Bayesian identification of electromechanical properties in piezoelectric energy harvesters, Mechanical Systems and Signal Processing 141 (2020), https://doi.org/10.1016/j.ymssp.2019.106506 106506.
[66] L. Lerimonti, I. Venanzi, N. Cavalagli, F. Comodini, F. Ubertini, An innovative continuous bayesian model updating method for base-isolated RC buildings using vibration monitoring data, Mechanical Systems and Signal Processing 139 (2020), https://doi.org/10.1016/j.ymssp.2019.106600 106600.
[67] S.H. Cheung, S. Bansal, A new Gibbs sampling based algorithm for Bayesian model updating with incomplete complex modal data, Mechanical Systems and Signal Processing 92 (2017) 156–172, https://doi.org/10.1016/j.ymssp.2017.01.015.
[68] T. Yin, H. Zhu, Selection of masters in dynamic reduction-based structural health monitoring using Bayesian experimental design, Mechanical Systems and Signal Processing 150 (2021), https://doi.org/10.1016/j.ymssp.2020.107294 107294.
[69] P. Gardner, C. Lord, R.J. Barthorpe, Bayesian history matching for structural dynamics applications, Mechanical Systems and Signal Processing 143 (2020), https://doi.org/10.1016/j.ymssp.2020.106828 106828.
[70] Z. Ding, J. Li, H. Hao, Structural damage identification using improved Jaya algorithm based on sparse regularization and Bayesian inference, Mechanical Systems and Signal Processing 132 (2019) 211–231, https://doi.org/10.1016/j.ymssp.2019.06.029.
[71] R. Rocchetta, M. Broggi, Q. Huchet, E. Patelli, On-line Bayesian model updating for structural health monitoring, Mechanical Systems and Signal Processing 103 (2018) 174–195, https://doi.org/10.1016/j.ymssp.2017.10.015.
[72] X.Y. Jia, 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, 2019, pp. 103–113, https://doi.org/10.7712/120219.6328.18902.
[73] X. Zhou, A. Montazeri, J.D. Albertson, Mobile sensing of point-source gas emissions using bayesian inference: An empirical examination of the likelihood function, Atmospheric Environment 218 (2019), https://doi.org/10.1016/j.atmosenv.2019.116981 116981.
[74] E. Yee, P.R. Kosteniuk, G.M. Chandler, C.A. Biltoft, J.F. Bowers, Statistical characteristics of concentration fluctuations in dispersing plumes in the atmospheric surface layer, Boundary-Layer Meteorology 65 (1993) 69–109, https://doi.org/10.1007/bf00708819.
[75] I. Senocak, N.W. Hengartner, M.B. Short, W.B. Daniel, Stochastic event reconstruction of atmospheric contaminant dispersion using Bayesian inference, Atmospheric Environment 42 (2008) 7718–7727, https://doi.org/10.1016/j.atmosenv.2008.05.024.
[76] D. Wade, I. Senocak, Stochastic reconstruction of multiple source atmospheric contaminant dispersion events, Atmospheric Environment 74 (2013) 45–51, https://doi.org/10.1016/j.atmosenv.2013.02.051.
[77] S. Bi, M. Broggi, M. Beer, The role of the Bhattacharyya distance in stochastic model updating, Mechanical Systems and Signal Processing 117 (2019) 437–452, https://doi.org/10.1016/j.ymssp.2018.08.017.
[78] A.B. Abdessalem, N. Dervilis, D. Wagg, K. Worden, Model selection and parameter estimation in structural dynamics using approximate Bayesian computation, Mechanical Systems and Signal Processing 99 (2018) 306–325, https://doi.org/10.1016/j.ymssp.2017.06.017.
[79] S. Fang, S. Chen, Y. Lin, Z. Dong, Probabilistic damage identification incorporating approximate Bayesian computation with stochastic response surface, Mechanical Systems and Signal Processing 128 (2019) 229–243, https://doi.org/10.1016/j.ymssp.2019.03.044.
[80] C. Scheidt, J. Caers, Uncertainty quantification in reservoir performance using distances and kernel methods – application to a West Africa deepwater turbidite reservoir, SPE Journal 14 (2009) 680–692, https://doi.org/10.2118/118740-pa.
[81] Reprint of: Mahalanobis, P.C. (1936) "On the Generalised Distance in Statistics", Sankhya A 80 (2018) 1–7. doi:10.1007/s13171-019-00164-5..
[82] A. Bhattacharyya, On a measure of divergence between two multinomial populations, The Indian Journal of Statistics 7 (1946) 401–406, https://doi.org/10.1038/157869b0.
[83] A. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vehtari, D.B. Rubin, Bayesian Data Analysis, 3rd ed., CRC Press, 2013, ISBN 978-1439898208.
[84] H. Raiffa, R. Schlaifer, Applied Statistical Decision Theory, Wiley, 2000, ISBN 978-0471383499.
[85] P. Diaconis, D. Ylvisaker, Conjugate priors for exponential families, The Annals of Statistics 7 (1979) 269–281, https://doi.org/10.1214/aos/1176344611.
[86] L.D. Brown, Fundamentals of Statistical Exponential Families: With Applications in Statistical Decision Theory, Institute of Mathematical Statistics (1986), ISBN: 978-0940600102.
[87] G. Casella, R.L. Berger, Statistical Inference, Brooks/Cole Publishing Company (1990), ISBN: 978-0534119584.
[88] T. Baldacchino, K. Worden, J. Rowson, Robust nonlinear system identification: Bayesian mixture of experts using the t-distribution, Mechanical Systems and Signal Processing 85 (2017) 977–992, https://doi.org/10.1016/j.ymssp.2016.08.045.
[89] M. Pharr, W. Jakob, G. Humphreys, Monte Carlo Integration, Physically Based Rendering (2017) 747–802, https://doi.org/10.1016/b978-0-12-800645-0.50013-0.
[90] R.L. Berger, Estimation: Point and interval, International Encyclopedia of the Social and Behavioral Sciences (2015) 16–20, https://doi.org/10.1016/b978-0-08-097086-8.42026-x.
[91] P.L. Green, 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 373 (2015), https://doi.org/10.1098/rsta.2014.0405, 20140405.
[92] P.E. Hadjidoukas, P. Angelikopoulos, D. Rossinelli, D. Alexeev, C. Papadimitriou, P. Koumoutsakos, Bayesian uncertainty quantification and propagation for discrete element simulations of granular materials, Computer Methods in Applied Mechanics and Engineering 282 (2014) 218–238, https://doi.org/10.1016/j.cma.2014.07.017.
[93] R. Rastogi, S. Ghosh, A.K. Ghosh, K.K. Vaze, P.K. Singh, Fatigue crack growth prediction in nuclear piping using Markov Chain Monte Carlo simulation, Fatigue and Fracture of Engineering Materials and Structures 40 (2016) 145–156, https://doi.org/10.1111/ffe.12486.
[94] S. Rouchier, M.J. Jimenez, S. Castano, Sequential Monte Carlo for on-line parameter estimation of a lumped building energy model, Energy and Buildings 187 (2019) 86–94, https://doi.org/10.1016/j.enbuild.2019.01.045.
[95] H. Jensen, D. Jerez, A Bayesian model updating approach for detection-related problems in water distribution networks, Reliability Engineering and System Safety 185 (2019) 100–112, https://doi.org/10.1016/j.ress.2018.12.014.
[96] D. Straub, I. Papaioannou, Bayesian updating with structural reliability methods, Journal of Engineering Mechanics 141 (2014), https://doi.org/10.1061/(ASCE)EM.1943-7889.0000839.
[97] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, Equation of state calculations by fast computing machines, The Journal of Chemical Physics 21 (1953) 1087–1092, https://doi.org/10.1063/1.1699114.
[98] P.A. Gagniuc, Markov chains: from theory to implementation and experimentation, West Sussex: Wiley Blackwell, Chichester, 2017, ISBN 978-1119387558.
[99] R. Serfozo, Basics of Applied Stochastic Processes, Springer, Berlin, Berlin, 2014, ISBN 978-3540893318.
[100] W.L. Dunn, J.K. Shultis, Exploring Monte Carlo Methods, Elsevier, 2011, ISBN 978-0444515759.
[101] J.K. Kruschke, Markov Chain Monte Carlo, Doing Bayesian Data Analysis (2015) 143–191, https://doi.org/10.1016/b978-0-12-405888-0.00007-6.
[102] W.K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, Biometrika 57 (1970) 97–109, https://doi.org/10.1093/biomet/57.1.97.

[103] A.E. Gelfand, A.F.M. Smith, Sampling-based approaches to calculating marginal densities, Journal of the American Statistical Association 85 (1990) 398–409, https://doi.org/10.1080/01621459.1990.10476213.
[104] W.R. Gilks, P. Wild, Adaptive rejection sampling for gibbs sampling, Applied Statistics 41 (1992) 337–348, https://doi.org/10.2307/2347565.
[105] S. Chib, Markov Chain Monte Carlo methods: computation and inference, Handbook of Econometrics 5 (2001) 3569–3649, https://doi.org/10.1016/s1573-4412(01)05010-3.
[106] X.S. Yang, Introduction to Algorithms for Data Mining and Machine Learning, 1st Edition., Academic Press, 2019, ISBN 978-0128172162.
[107] S. Brooks, A. Gelman, G. Jones, X.L. Meng, Handbook of Markov Chain Monte Carlo, CRC Press, 2011, ISBN 978-1420079425.
[108] D.J. Wilkinson, Parallel Bayesian computation, Handbook of Parallel Computing and Statistics (2005) 481–512.
[109] A.E. Brockwell, Parallel Markov Chain Monte Carlo simulation by pre-fetching, Journal of Computational and Graphical Statistics 15 (2006) 246–261, https://doi.org/10.1198/106186006x100579.
[110] R.M. Neal, Slice sampling, The Annals of Statistics 31 (2003) 705–767, https://doi.org/10.1214/aos/1056562461.
[111] S. Duane, A. Kennedy, B.J. Pendleton, D. Roweth, Hybrid Monte Carlo, Physics Letters B 195 (1987) 216–222, https://doi.org/10.1016/0370-2693(87)91197-x.
[112] M. Girolami, B. Calderhead, Riemann manifold Langevin and Hamiltonian Monte Carlo methods, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73 (2011) 123–214, https://doi.org/10.1111/j.1467-9868.2010.00765.x.
[113] 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 56 (1994) 591–592.
[114] G.O. Roberts, R.L. Tweedie, Exponential convergence of langevin distributions and their discrete approximations, Bernoulli 2 (1996) 341, https://doi.org/10.2307/3318418.
[115] G.O. Roberts, J.S. Rosenthal, Optimal scaling of discrete approximations to Langevin diffusions, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 60 (1998) 255–268, https://doi.org/10.1111/1467-9868.00123.
[116] J.S. Liu, F. Liang, W.H. Wong, The multiple-try method and local optimization in metropolis sampling, Journal of the American Statistical Association 95 (2000) 121–134, https://doi.org/10.1080/01621459.2000.10473908.
[117] L. Martino, A review of multiple try MCMC algorithms for signal processing, Digital Signal Processing 75 (2018) 134–152, https://doi.org/10.1016/j.dsp.2018.01.004.
[118] P.J. Green, Reversible jump Markov chain Monte Carlo computation and Bayesian model determination, Biometrika 82 (1995) 711–732, https://doi.org/10.1093/biomet/82.4.711.
[119] C. Andrieu, G.O. Roberts, The pseudo-marginal approach for efficient Monte Carlo computations, The Annals of Statistics 37 (2009) 697–725, https://doi.org/10.1214/07-aos574.
[120] G.O. Roberts, J.S. Rosenthal, Optimal scaling for various metropolis-hastings algorithms, Statistical Science 16 (2001) 351–367, https://doi.org/10.1214/ss/1015346320.
[121] G.O. Roberts, A. Gelman, W.R. Gilks, Weak convergence and optimal scaling of random walk metropolis algorithms, The Annals of Applied Probability 7 (1997) 110–120, https://doi.org/10.1214/aoap/1034625254.
[122] M. Bedard, Optimal acceptance rates for metropolis algorithms: Moving beyond 0.234, Stochastic Processes and Their Applications 118 (2008) 2198–2222. doi:10.1016/j.spa.2007.12.005..
[123] S. Chib, E. Greenberg, Understanding the metropolis-hastings algorithm, Journal of Engineering Mechanics 49 (1995) 327, https://doi.org/10.2307/2684568.
[124] R. Costilla, I. Liu, R. Arnold, D. Fernandez, Bayesian model-based clustering for longitudinal ordinal data, Computational Statistics 34 (2019) 1015–1038, https://doi.org/10.1007/s00180-019-00872-4.
[125] A.A. Johnson, G.L. Jones, R.C. Neath, Component-wise Markov Chain Monte Carlo: Uniform and geometric ergodicity under mixing and composition, Statistical Science 28 (2013) 360–375, https://doi.org/10.1214/13-sts423.
[126] H. Haario, E. Saksman, J. Tamminen, Componentwise adaptation for high dimensional MCMC, Computational Statistics 20 (2005) 265–273, https://doi.org/10.1007/bf02789703.
[127] R.A. Levine, Z. Yu, W.G. Hanley, J.J. Nitao, Implementing componentwise Hastings algorithms, Computational Statistics and Data Analysis 48 (2005) 363–389, https://doi.org/10.1016/j.csda.2004.02.002.
[128] G. Casella, E.I. George, Explaining the gibbs sampler, The American Statistician 46 (1992) 167–174, https://doi.org/10.2307/2685208.
[129] P. Honarmandi, L. Johnson, R. Arroyave, Bayesian probabilistic prediction of precipitation behavior in Ni-Ti shape memory alloys, Computational Materials Science 172 (2020), https://doi.org/10.1016/j.commatsci.2019.109334 109334.
[130] K. Kawahara, R. Ishikawa, T. Higashi, T. Kimura, Y.H. Ikuhara, N. Shibata, Y. Ikuhara, Unique fitting of electrochemical impedance spectra by random walk Metropolis Hastings algorithm, Journal of Power Sources 403 (2018) 184–191, https://doi.org/10.1016/j.jpowsour.2018.09.091.
[131] H. Wang, C. Wang, Y. Wang, X. Gao, C. Yu, Bayesian forecasting and uncertainty quantifying of stream flows using Metropolis-Hastings Markov Chain Monte Carlo algorithm, Journal of Hydrology 549 (2017) 476–483, https://doi.org/10.1016/j.jhydrol.2017.03.073.
[132] P. Grišins, I.E. Mazets, Metropolis-Hastings thermal state sampling for numerical simulations of Bose-Einstein condensates, Computer Physics Communications 185 (2014) 1926–1931, https://doi.org/10.1016/j.cpc.2014.03.021.
[133] S. Biswal, A. Ramaswamy, Finite element model updating of concrete structures based on imprecise probability, Mechanical Systems and Signal Processing 94 (2017) 165–179, https://doi.org/10.1016/j.ymssp.2017.02.042.
[134] P. Green, E. Cross, K. Worden, Bayesian system identification of dynamical systems using highly informative training data, Mechanical Systems and Signal Processing 56–57 (2015) 109–122, https://doi.org/10.1016/j.ymssp.2014.10.003.
[135] J.L. Beck, S.K. Au, Bayesian updating of structural models and reliability using Markov chain Monte Carlo simulation, Journal of Engineering Mechanics 128 (2002) 380–391, https://doi.org/10.1061/(ASCE)0733-9399(2002)128:4(380).
[136] W. Betz, I. Papaioannou, D. Straub, Transitional Markov chain Monte Carlo: observations and improvements, Journal of Engineering Mechanics 142 (2016), https://doi.org/10.1061/(asce)em.1943-7889.0001066.
[137] P. Efraimidis, P. Spirakis, Weighted random sampling, Encyclopedia of Algorithms (2008) 1024–1027, https://doi.org/10.1007/978-0-387-30162-4_478.
[138] J.Y. Ching, J.S. Wang, Application of the transitional Markov chain Monte Carlo algorithm to probabilistic site characterization, Engineering Geology 203 (2016) 151–167, https://doi.org/10.1016/j.enggeo.2015.10.015.
[139] G.A. Ortiz, D.A. Alvarez, D. Bedoya-Ruíz, Identification of Bouc–wen type models using the transitional Markov chain Monte Carlo method, Computers and Structures 146 (2015) 252–269, https://doi.org/10.1016/j.compstruc.2014.10.012.
[140] W.H. Zhou, F. Tan, K.V. Yuen, Model updating and uncertainty analysis for creep behavior of soft soil, Computers and Geotechnics 100 (2018) 135–143, https://doi.org/10.1016/j.compgeo.2018.04.006.
[141] J. Wang, L. Katafygiotis, Reliability-based optimal design of linear structures subjected to stochastic excitations, Structural Safety 47 (2014) 29–38, https://doi.org/10.1016/j.strusafe.2013.11.002.
[142] H.M. Gomes, M. Broggi, E. Patelli, J.E. Mottershead, Model Updating by Uncertain Parameter Inference, Vulnerability, Uncertainty, and Risk (2014), https://doi.org/10.1061/9780784413609.153.
[143] O. Sedehi, C. Papadimitriou, L.S. Katafygiotis, Probabilistic hierarchical bayesian framework for time-domain model updating and robust predictions, Mechanical Systems and Signal Processing 123 (2019) 648–673, https://doi.org/10.1016/j.ymssp.2018.09.041.
[144] N.J. Gordon, D.J. Salmond, A.F.M. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation, IEE Proceedings F (Radar and Signal Processing) 140 (1993) 107, https://doi.org/10.1049/ip-f-2.1993.0015.

[145] P.D. Moral, Nonlinear filtering: Interacting particle resolution, Comptes Rendus De LAcadémie Des Sciences – Series I – Mathematics 325 (1997) 653–658, https://doi.org/10.1016/s0764-4442(97)84778-7.
[146] J.S. Liu, R. Chen, Sequential Monte Carlo methods for dynamic systems, Journal of the American Statistical Association 93 (1998) 1032–1044, https://doi.org/10.1080/01621459.1998.10473765.
[147] S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for on-line nonlinear/non-gaussian bayesian tracking, IEEE Transactions on Signal Processing 50 (2002) 174–188, https://doi.org/10.1109/78.978374.
[148] B. Ristic, S. Arulampalam, N. Gordon, Beyond the Kalman Filter: Particle Filters for Tracking Applications, Artech House, Boston, 2004, ISBN 978-1580538510.
[149] A. Doucet, N. de Freitas, N. Gordon, A. Smith, Sequential Monte Carlo Methods in Practice, Springer, New York, 2001, ISBN 978-0387951461.
[150] O. Cappe, S.J. Godsill, E. Moulines, An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo, Proceedings of the IEEE 95 (2007) 899–924, https://doi.org/10.1109/jproc.2007.893250.
[151] T.J. Rogers, K. Worden, E.J. Cross, Bayesian Solutions to State-Space Structural Identification, Model Validation and Uncertainty Quantification 3 (2020) 247–253, https://doi.org/10.1007/978-3-030-47638-0_27.
[152] T.J. Rogers, T.B. Schon, A. Lindholm, K. Worden, E.J. Cross, Identification of a Duffing oscillator using particle Gibbs with ancestor sampling, Journal of Physics: Conference Series 1264 (2019), https://doi.org/10.1088/1742-6596/1264/1/012051 012051.
[153] P.D. Moral, A. Doucet, A. Jasra, Sequential Monte Carlo samplers, Journal of the Royal Statistical Society. Series B (Statistical Methodology) 68 (2006) 411–436.
[154] M. Beer, I.A. Kougioumtzoglou, E. Patelli, Emerging Concepts and Approaches for Efficient and Realistic Uncertainty Quantification, Maintenance and Safety of Aging Infrastructure (2014) 121–162, https://doi.org/10.1201/b17073-5.
[155] A. Doucet, M. Briers, S. Sénécal, Efficient block sampling strategies for sequential Monte Carlo methods, Journal of Computational and Graphical Statistics 15 (2006) 693–711, https://doi.org/10.1198/106186006x142744.
[156] P.D. Moral, A. Doucet, A. Jasra, On adaptive resampling strategies for sequential Monte Carlo methods, Bernoulli 18 (2012) 252–278, https://doi.org/10.3150/10-bej335.
[157] A. Jasra, D.A. Stephens, A. Doucet, T. Tsagaris, Inference for Lévy-Driven stochastic volatility models via adaptive sequential Monte Carlo, Scandinavian Journal of Statistics 38 (2010) 1–22, https://doi.org/10.1111/j.1467-9469.2010.00723.x.
[158] C.A. Naesseth, F. Lindsten, T.B. Schön, Nested Sequential Monte Carlo Methods, Proceedings of the 32nd International Conference on International Conference on Machine Learning 37 (2015) 1292–1301..
[159] C.A. Naesseth, F. Lindsten, T.B. Schön, High-dimensional filtering using nested sequential Monte Carlo, IEEE Transactions on Signal Processing 67 (2019) 4177–4188, https://doi.org/10.1109/TSP.2019.2926035.
[160] J.M. Hammersley, D.C. Handscomb, Monte Carlo Methods, Springer, Netherlands, 1964, ISBN 978-9400958197.
[161] R.Y. Rubinstein, D.P. Kroese, Simulation and the Monte Carlo Method, 3rd Edition., John Wiley & Sons, 2016, ISBN 978-1118632161.
[162] S. Theodoridis, Monte Carlo Methods, Machine Learning (2015) 707–744, https://doi.org/10.1016/b978-0-12-801522-3.00014-8.
[163] A. Kong, J.S. Liu, W.H. Wong, Sequential imputations and Bayesian missing data problems, Journal of the American Statistical Association 89 (1994) 278–288, https://doi.org/10.1080/01621459.1994.10476469.
[164] K.H. Li, The sampling/importance resampling algorithm, applied Bayesian modeling and causal inference from incomplete-data, Perspectives (2004) 265–276, https://doi.org/10.1002/0470090456.ch24.
[165] M. West, Approximating posterior distributions by mixtures, Journal of the Royal Statistical Society: Series B (Methodological) 55 (1993) 409–422, https://doi.org/10.1111/j.2517-6161.1993.tb01911.x.
[166] N. Chopin, A sequential particle filter method for static models, Biometrika 89 (2002) 539–552, https://doi.org/10.1093/biomet/89.3.539.
[167] D. Basaran, A.T. Cemgil, E. Anarim, Multiresolution alignment for multiple unsynchronized audio sequences using sequential Monte Carlo samplers, Digital Signal Processing 77 (2018) 77–85, https://doi.org/10.1016/j.dsp.2017.10.024.
[168] T.B. Schön, A. Svensson, L. Murray, F. Lindsten, Probabilistic Learning of Nonlinear Dynamical Systems using Sequential Monte Carlo Mechanical Systems and Signal Processing 104 (2018) 866–883, https://doi.org/10.1016/j.ymssp.2017.10.033.
[169] G. Zhu, X. Li, J. Ma, Y. Wang, S. Liu, C. Huang, K. Zhang, X. Hu, A new moving strategy for the Sequential Monte Carlo approach in optimizing the hydrological model parameters, Advances in Water Resources 114 (2018) 164–179, https://doi.org/10.1016/j.advwatres.2018.02.007.
[170] J. Sun, H. Zuo, W. Wang, 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 45 (2014) 396–407, https://doi.org/10.1016/j.ymssp.2013.08.022.
[171] A. Svensson, T.B. Schon, F. Lindsten, Learning of state-space models with highly informative observations: A tempered sequential Monte Carlo solution, Mechanical Systems and Signal Processing 104 (2018) 915–928, https://doi.org/10.1016/j.ymssp.2017.09.016.
[172] D. Wang, S. Sun, 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 52–53 (2015) 293–308, https://doi.org/10.1016/j.ymssp.2014.07.005.
[173] J. Wolberg, Data analysis using the method of least squares extracting the most information from experiments, Springer, Berlin Heidelberg, Berlin, Heidelberg, 2006, ISBN 978-3540317203.
[174] A.Y. Pisal, R.S. Jangid, Dynamic response of structure with tuned mass friction damper, International Journal of Advanced Structural Engineering 8 (2016) 363–377, https://doi.org/10.1007/s40091-016-0136-7.
[175] J.L. Beck, S.K. Au, Bayesian updating of structural models and reliability using Markov Chain Monte Carlo simulation, Journal of Engineering Mechanics 128 (2002) 380–391, https://doi.org/10.1061/(asce)0733-9399(2002)128:4(380).
[176] 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, 1997, pp. 1346–1352.
[177] Y. Govers, H.H. Khodaparast, M. Link, J. Mottershead, A comparison of two stochastic model updating methods using the DLR-AIRMOD test structure, Mechanical Systems and Signal Processing 52–53 (2015) 105–114, https://doi.org/10.1016/j.ymssp.2014.06.003.
[178] M. Broggi, M. Faes, E. Patelli, Y. Govers, D. Moens, M. Beer, Comparison of Bayesian and interval uncertainty quantification: Application to the AIRMOD test structure, 2017 IEEE Symposium Series on Computational Intelligence (SSCI) (2017), https://doi.org/10.1109/ssci.2017.8280882.
[179] E. Patelli, Y. Govers, M. Broggi, H.M. Gomes, M. Link, J.E. Mottershead, Sensitivity or Bayesian model updating: A comparison of techniques using the DLR-AIRMOD test data, Archive of Applied Mechanics 87 (2017) 905–925, https://doi.org/10.1007/s00419-017-1233-1.
[180] J.E. Mottershead, M. Link, M.I. Friswell, The sensitivity method in finite element model updating: A tutorial, Mechanical Systems and Signal Processing 25 (2011) 2275–2296, https://doi.org/10.1016/j.ymssp.2010.10.012.
[181] J.E. Mottershead, M.I. Friswell, Model updating in structural dynamics: A survey, Journal of Sound and Vibration 167 (1993) 347–375, https://doi.org/10.1006/jsvi.1993.1340.
[182] E. Patelli, COSSAN: A Multidisciplinary Software Suite for Uncertainty Quantification and Risk Management, Handbook of Uncertainty Quantification (2017) 1909–1977, https://doi.org/10.1007/978-3-319-12385-1_59.
[183] E. Patelli, M. Broggi, M.D. Angelis, M. Beer, OpenCossan: An Efficient Open Tool for Dealing with Epistemic and Aleatory Uncertainties, Vulnerability, Uncertainty, and Risk (2014), https://doi.org/10.1061/9780784413609.258.
[184] F. Massey, The Kolmogorov-Smirnov test for goodness of fit, Journal of the American Statistical Association 46 (1951) 68–78, https://doi.org/10.2307/2280095.
[185] G. Marsaglia, W.W. Tsang, J. Wang, Evaluating Kolmogorovs Distribution, Journal of Statistical Software 8. doi:10.18637/jss.v008.i18..
[186] M.A. Stephens, Introduction to Kolmogorov (1933) On the Empirical Determination of a Distribution, Springer Series in Statistics Breakthroughs in Statistics (1992) 93–105. doi:10.1007/978-1-4612-4380-9_9..
[187] S.K. Au, J.L. Beck, Important sampling in high dimensions, Structural Safety 25 (2003) 139–163, https://doi.org/10.1016/s0167-4730(02)00047-4.

[188] C. Snyder, T. Bengtsson, P. Bickel, J. Anderson, Obstacles to high-dimensional particle filtering, Monthly Weather Review 136 (2008) 4629–4640, https://doi.org/10.1175/2008mwr2529.1.
[189] P. Rebeschini, R.V. Handel, Can local particle filters beat the curse of dimensionality?, The Annals of Applied Probability 25 (2015) 2809–2866, https://doi.org/10.1214/14-aap1061.
[190] G.O. Roberts, J.S. Rosenthal, Coupling and ergodicity of adaptive Markov Chain Monte Carlo Algorithms, Journal of Applied Probability 44 (2007) 458–475, https://doi.org/10.1239/jap/1183667414.
[191] G.O. Roberts, J.S. Rosenthal, Examples of Adaptive MCMC, Journal of Computational and Graphical Statistics 18 (2009) 349–367, https://doi.org/10.1198/jcgs.2009.06134.
[192] H.F. Lam, J. Yang, 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 102 (2015) 144–155, https://doi.org/10.1016/j.engstruct.2015.08.005.
[193] N. Kantas, A. Beskos, 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 2 (2014) 464–489, https://doi.org/10.1137/130930364.
[194] H.D. Vo, Z. Fox, A. Baetica, B. Munsky, Bayesian estimation for stochastic gene expression using multifidelity models, The Journal of Physical Chemistry B 123 (2019) 2217–2234, https://doi.org/10.1021/acs.jpcb.8b10946.
[195] M.N. Chatzis, E.N. Chatzi, Online Bayesian identification of non-smooth systems, Procedia Engineering 199 (2017) 918–923, https://doi.org/10.1016/j.proeng.2017.09.238.
[196] J. Nakajima, M. Kasuya, 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 25 (2011) 225–245, https://doi.org/10.1016/j.jjie.2011.07.004.
[197] S. Dimitrakopoulos, Semiparametric bayesian inference for time-varying parameter regression models with stochastic volatility, Economics Letters 150 (2017) 10–14, https://doi.org/10.1016/j.econlet.2016.10.035.
[198] S. Mallapur, R. Platz, Uncertainty quantification in the mathematical modelling of a suspension strut using Bayesian inference, Mechanical Systems and Signal Processing 118 (2019) 158–170, https://doi.org/10.1016/j.ymssp.2018.08.046.
[199] R. Fuentes, N. Dervilis, K. Worden, E. Cross, Efficient parameter identification and model selection in nonlinear dynamical systems via sparse Bayesian learning, Journal of Physics: Conference Series 1264 (2019), https://doi.org/10.1088/1742-6596/1264/1/012050 012050.