Available online at www.eccomasproceedia.org
Eccomas Proceedia COMPDYN (2023) 2348-2368

ECCOMAS
Proceedia

COMPDYN 2023
9th ECCOMAS Thematic Conference on
Computational Methods in Structural Dynamics and Earthquake Engineering
M. Papadrakakis, M. Fragiadakis (eds.)
Athens, Greece, 12-14 June 2023

A COMPARISON OF SAMPLING-BASED BAYESIAN MODEL UPDATING APPROACHES APPLIED UPON A NEW BENCHMARK AEROSPACE TESTBED

Ewan B. Smith[1], **Kui He**[2], **and Sifeng Bi**[1]

[1]Aerospace Centre of Excellence
James Weir Building, University of Strathclyde, Glasgow, G1 1XJ, United Kingdom
{ewan.smith, sifeng.bi}@strath.ac.uk

[2] School of Aerospace Engineering
Beijing Institute of Technology, Beijing, 100081, China
kui.he@bit.edu.cn

Abstract. *Efforts to correlate numerical models to the results of experimental data requires the use of model updating techniques. While deterministic methods are well established, new developments in stochastic methods seek to update models when uncertainties in inputs and outputs are observed. In Bayesian frameworks, many sampling methods have been proposed such as TMCMC, iTMCMC, BUS and aBUS, all of which extend classical MCMC methods. In this work, these methods are tested and compared when updating both a simulated 3 degree-of-freedom spring-mass system and an up and coming experimental stochastic model updating benchmark. Results reveal that TMCMC provides the most accurate calibration with acceptable efficiency, although MCMC is highly efficient if some inaccuracy is allowable. The effectiveness of the improvements brought by iTMCMC compared to standard TMCMC, as well as the performance of BUS/aBUS, appear to depend heavily upon the case at hand.*

Keywords: Bayesian model updating, aerospace structural dynamics

ISSN:2623-3347 © 2023 The Authors. Published by Eccomas Proceedia.
Peer-review under responsibility of the organizing committee of COMPDYN 2023.
doi: 10.7712/120123.10565.21018

1 INTRODUCTION

Model updating has established itself as a key strategy within the field of engineering for the correction of numerical models such that they match outputs found in experimental measurements. In the deterministic sense, strategies such as maximum-likelihood estimation (MLE) [1, 2] or sensitivity analysis-based updating [3, 4] are popular and have proven themselves as capable of updating models of varying complexity with suitable efficiency and accuracy. In practice, however, there are approximations in numerical simulations as well as unavoidable uncertainties in experiments which require careful consideration when performing model calibration. Such existence of uncertainties within a mechanical system's inputs and/or outputs has shifted model updating focus from deterministic approaches to stochastic approaches [5]. Typically performed via a Bayesian inference framework, stochastic model updating approximates the posterior probability of parameter values from *a priori* knowledge or beliefs. In such frameworks, core techniques lay on attempts to both sample from and approximate an unknown posterior.

Across disciplines, Markov-chain Monte Carlo (MCMC) [6] has been classically employed to perform Bayesian inference. However, as the complexity of model updating problems has grown, a realisation has come that MCMC is not only ineffective at the sampling of complex posteriors, but is also inefficient with highly dimensional problems. This has led to the development of alternative sampling methods, such as the popular Transitional Markov-chain Monte Carlo (TMCMC) [7]. In literature, one can find an abundance of such methods, all with varying levels of tuning and varying levels of performance relative to the complexity of the problem at hand.

Here, we investigate some of the most prominent sampling methods in discuss in engineering contexts: Markov-chain Monte Carlo (MCMC), Transitional Markov-chain Monte Carlo (TMCMC), Improved Transitional Markov-chain Monte Carlo (iTMCMC) [8], and Bayesian Updating with Structural Methods (BUS) [9], including its so-called adaptive version (aBUS) [10]. To enable comparison, we employ each method to first update a simple test case - a 3 degree of freedom spring-mass system. In this case, simulated 'experimental' outputs are generated by running the model with known true values and the model is then calibrated to these outputs. To evaluate a more complex case, the methods are then used to update a new in-development stochastic model updating benchmark testbed based upon a small scale airplane and involving multi-source uncertainties and with measurements from laboratory experiments. This updating case involves the calibration of a Finite Element (FE) model which aims to replicate the experiment.

The various sampling-based updating methods will be compared quantitatively upon their ability to update model, ensuring proper exploration of posterior probability density functions (PDFs) with suitable accuracy and robustness. As structural FE simulations are typically costly computationally, we seek sampling methods which minimize the amount of model runs required for approximation, without degrading the accuracy of resultant statistical estimators.

2 METHODS AND MATERIALS

2.1 Bayesian model updating framework

Consider an engineering system which can be characterized by the following:

$$\mathbf{y} = h(\boldsymbol{\theta}) \tag{1}$$

where $\mathbf{y} = [y_1, y_2, ..., y_m]$ is a vector of m measured outputs, h is a numerical model or physical system, and $\boldsymbol{\theta} = [\theta_1, \theta_2, ..., \theta_n]$ is a vector of n input system parameters. When comparing the outputs of an experiment and a numerical model of the same system, $\mathbf{y}_{exp}$ and $\mathbf{y}_{\mathbf{sim}}$ respectively, there is likely a residual (error), ϵ, between them, even for the same set of input parameters θ:

$$\mathbf{y}_{\mathbf{sim}} = \mathbf{y}_{exp} + \epsilon \tag{2}$$

Differences between experimental and model outputs can stem from parameter uncertainty (unknown input parameters and boundary conditions), modelling error (inherent approximations in models), and/or experimental error (human errors in measurements). Traditional deterministic model updating seeks to reduce this residual error by updating model inputs such that model outputs match experimental measurements [4]. In its simplest form, deterministic model updating is an optimization problem wherein the residual is minimized. Because of it's reduction to an optimization problem, model updating has been successfully performed with a variety of traditional techniques such as genetic algorithms [11], particle swarms [12], simulated annealing [13] and harmony search [14].

In most structural applications however, there exists unavoidable aleatoric and epistemic uncertainties in model inputs which propagate from sources such as measurement errors or assumptions/simplifications in modelling. Furthermore, it may be that structural parameters or the outputs themselves are non-stationary and may vary with repeated tests. Such notions have brought developments into stochastic model updating.

The most popular approach to stochastic model updating is Bayesian inference [15], which carries the advantage of allowing for prior knowledge to be incorporated into posterior probability estimation. Considering a case where $\mathbf{y}_{exp}$ is a vector of experimental measurements and $\boldsymbol{\theta}$ is a vector of input parameters, Bayes' theorem can be applied as:

$$P(\boldsymbol{\theta}|\mathbf{y}_{exp}) = \frac{P(\mathbf{y}_{exp}|\boldsymbol{\theta}) \cdot P(\boldsymbol{\theta})}{P(\mathbf{y}_{exp})} \tag{3}$$

where $P(\boldsymbol{\theta}|\mathbf{y}_{exp})$ is the posterior distribution, $P(\mathbf{y}_{exp}|\theta)$ is the likelihood function which measures the agreement between experimental and numerical results, $P(\boldsymbol{\theta})$ is the prior distribution, and $P(\mathbf{y}_{exp})$ is the evidence, or, the probability of producing the data. The evidence acts as a normalising constant to ensure that the posterior distribution integrates to 1 and is completely independent of the parameters $\boldsymbol{\theta}$. In practice, the calculation of the evidence is tricky, especially in large dimensions. As such, if only the relationship between the parameters $\boldsymbol{\theta}$ and the observations $\mathbf{y}_{exp}$ is of interest the evidence can be neglected and the posterior is henceforth proportionally related to the prior and likelihood:

$$P(\boldsymbol{\theta}|\mathbf{y}_{exp}) \propto P(\mathbf{y}_{exp}|\boldsymbol{\theta}) \cdot P(\boldsymbol{\theta}) \tag{4}$$

In most cases, the analytical derivation of a likelihood function requires a massive amount of model evaluation or may even be intractable. Fortunately, Approximate Bayesian Computation (ABC) [16, 17] allows for the likelihood to be replaced with an efficient yet still accurate function to measure model data's closeness to the experimental data. Finally, the likelihood for each distance metric is defined as:

$$P_L(\mathbf{y}_{exp}|\boldsymbol{\theta}) \propto \exp\left\{-\frac{s^2}{\varepsilon^2}\right\} \tag{5}$$

where s is some sort of statistic which measures the closeness of data and ε is a ”width factor”, the value of which has the potential to tune the convergence performance of model updating. Smaller values of ε creates a posterior which is peaked and is more likely to predict the true value, at the expense of extra computation.

To analyse stochastic models, uncertainties of the system need to first be characterized and propagated through the system to generate uncertain outputs, presenting themselves as probability distributions, intervals, or fuzzy number sets. Generally, uncertainty is characterised as uncertain input parameters and propagation is achieved by Monte Carlo simulation where input parameters are varied across a user-determined amount of samples. Regardless of the form of uncertainties, uncertainty quantification algorithms make use of the random samples to generate their statistics. Perhaps the most simple UQ metric is the Euclidean distance:

$$d_{euc}(\mathbf{y}_{exp}, \mathbf{y}_{sim}) = \sqrt{(\bar{\mathbf{y}}_{exp} - \bar{\mathbf{y}}_{sim})(\bar{\mathbf{y}}_{exp} - \bar{\mathbf{y}}_{sim})^T} \tag{6}$$

where $\bar{\mathbf{y}}_{exp}$ and $\bar{\mathbf{y}}_{sim}$ are the row vectors of the means of the experimental and simulation data, respectively. While the Euclidean distance successfully captures the mean values of the datasets and can successfully update mean parameters, it is fruitful to use a UQ metric which captures not only the mean but also variance and even the shape of distributions, such as the Bhattacharyya distance [18]:

$$d_{bha}(\mathbf{y}_{exp}, \mathbf{y}_{sim}) = -\log\left[\int_{\mathbb{X}} \sqrt{p_{exp}(x)p_{sim}(x)}dx\right] \tag{7}$$

where $p_{exp}(x)$ and $p_{sim}(x)$ are the probability density functions (PDFs) of the experimental and simulated data, and $\mathbb{X}$ is the m-dimensional sample space. Proposed for use in model updating by Bi et al. [19], the Bhattacharyya distance can be discretised by means of binning:

$$d_{bha}(\mathbf{y}_{exp}, \mathbf{y}_{sim}) = -\log\left\{\sum_{i_m=1}^{n_{bin}} \sum_{i_1=1}^{n_{bin}} \sqrt{p_{exp}(b_{i_1,i_2,\ldots,i_m})p_{sim}(b_{i_1,i_2,\ldots,i_m})}\right\} \tag{8}$$

where $p_{exp}(b_{i_1,i_2,\ldots,i_m})$ and $p_{sim}(b_{i_1,i_2,\ldots,i_m})$ denote the probability mass function (PMF) value for the bin $b_{i_1,i_2,\ldots,i_m}$. The number of bins is congruent to the number of output samples for experiment and simulation:

$$n_{bin} \cong \left[\frac{\max(N_{sim}, N_{exp})}{10}\right] \tag{9}$$

With this, the range between the minimum and maximum values of $\mathbf{y}$ can be split into n_{bin} intervals and the probability mass of the bin can be calculated as:

$$p_{sim}(b_{a_i,b_i}) = \frac{\text{number of points in bin between } a \text{ and } b}{N_{sim}} \tag{10}$$

$$p_{exp}(b_{a_i,b_i}) = \frac{\text{number of points in bin between } a \text{ and } b}{N_{exp}} \tag{11}$$

Hence, with all of this, the Euclidean and Bhattacharyya distance-based approximate likelihoods can be defined by:

$$P_L(\mathbf{y}_{exp}|\boldsymbol{\theta}) \propto \exp\left\{-\frac{d^2}{\varepsilon^2}\right\} \tag{12}$$

where d is either the Euclidean or Bhattacharyya distance metric and ε is a 'width' parameter to control how 'peaked' the likelihood function is. Smaller values of ε lead to posteriors which are more likely to converge to true values but require larger computation. In testing, a width value of $\sim$0.01 and $\sim$0.1 were found to be optimal for the 3 degree of freedom spring-mass system and airplane test cases, respectively.

Similar to Bi et al. [19], The Euclidean and Bhattacharyya distances are combined into a 'two-step' updating strategy where the parameters reflecting the mean value of model inputs are first updated by the Euclidean distance, and then variance parameters are updated by the Bhattacharyya distance. As part of this strategy, the posterior distributions obtained by Euclidean-based updating are used as prior distributions in Bhattacharyya-based updated to save computational effort as the calculation of Bhattacharyya distances is computationally more taxing.

2.2 Markov-chain Monte Carlo (MCMC)

Markov Chain Monte Carlo (MCMC) is a statistical sampling method which allows for the characterisation of a probability distribution without explicit knowledge of the distribution's specific properties. The method is a combination of Monte Carlo simulation and Markov chain processes, where subsequent Monte Carlo samples are taken relative to the previous sample's location within the domain. In practice, this means that the next proposed sample, θ^*, is randomly drawn from a proposal distribution, such as a Normal distribution, which has its mean centred around the previous sample's value. Hence, this creates a long Markov chain of randomly generated samples with each sample's probability being solely dependant upon location within the parameter space of only the previous sample.

For MCMC to be useful within model updating, it needs an additional acceptance/rejection criteria to allow for the acceptance or rejection of generated samples, with the accepted samples then usable for an estimation of the posterior distribution. Without such criteria, the MCMC algorithm will simply generate a 'random walk' with no distinct convergence. The most popular acceptance criterion is the Metropolis-Hastings (MH) algorithm [20]. The MH algorithm defines a probability that a proposed sample will be accepted over the existing sample:

$$\alpha = min\left[1, \frac{P(\theta^*|\mathbf{D})}{P(\theta_i|\mathbf{D})} \cdot \frac{q(\theta_i|\theta^*)}{q(\theta^*|\theta_i)}\right] \tag{13}$$

where $P(\theta^*|\mathbf{D})$ is the posterior probability of the proposal sample, $P(\theta_i|\mathbf{D})$ is the posterior probability of the current sample, $q(\theta_i|\theta^*)$ is the probability of sampling θ_i given θ^* is the current sample and $q(\theta^*|\theta_i)$ is the probability of sampling θ^* given θ_i is the current sample. Substituting in Bayes' theorem (Eq. 3):

$$\alpha = min\left[1, \frac{P(\mathbf{D}|\theta^*)}{P(\mathbf{D}|\theta_i)} \cdot \frac{P(\theta^*)/P(\mathbf{D})}{P(\theta_i)/P(\mathbf{D})} \cdot \frac{q(\theta_i|\theta^*)}{q(\theta^*|\theta_i)}\right] \tag{14}$$

Because proposal distributions are typically symmetrical, the probability of sampling both the current and proposal samples is the same ($q(\theta_i|\theta^*) = q(\theta^*|\theta_i)$) and thus cancels out along with the evidence, $P(\mathbf{D})$:

$$\alpha = min\left[1, \frac{P(\theta^*|\mathbf{D})}{P(\theta_i|\mathbf{D})}\right] \tag{15}$$

MCMC has several user-tunable parameters. Perhaps most critically, is the choice and subsequent statistical parameters of the proposal distribution which has the ability to strongly influence the efficiency of sampling. A wide proposal PDF will explore the posterior but have a potentially low acceptance rate, while a narrow proposal PDF will have a high acceptance rate but with limited exploration of the posterior and high correlation between samples. A typical choice for a proposal distribution is a normal (or multi-variate normal in multi-dimensional problems) distribution where the mean is centred around the sample and which has a variation that can be tuned for optimizing sampler performance.

It is advisable when using MCMC to introduce a 'burn-in' period - an N_b number of samples to discard from the start of the chain - in cases where starting samples are far from the posterior. By discarding unneeded samples the accuracy of subsequent statistical estimations is improved. Figure 1 shows a visual example of burnin with MCMC.

2.3 Transitional Markov-chain Monte Carlo (TMCMC)

MCMC struggles to sample from more complex posteriors: multi-modal, highly peaked, or flat manifold distributions can see the chain getting 'stuck' within peaks and not fully exploring the space of the posterior. A variation of MCMC sampling proposed by Ching and Chen named Transitional Markov Chain Monte Carlo (TMCMC) aims to overcome this drawback by drawing samples from several intermediate distributions, generated iteratively, which then converge towards the complicated PDF. These constituent PDFs are defined as such:

$$P^j \propto P(\mathbf{D}|\boldsymbol{\theta})^{\beta_j} \cdot P(\boldsymbol{\theta}) \tag{16}$$

where j indicates the iteration count increasing from 0 to m (m being the total number of iterations), and β_j is the so-called tempering parameter which evolves from $\beta_0 = 0$ to $\beta_m = 1$. The purpose of this parameter is to moderate the evolution of a prior distribution to the posterior distribution (i.e. $P^0 = P(\boldsymbol{\theta})$ to $P^m = P(\boldsymbol{\theta}|\mathbf{D})$) such that the change from prior to posterior is gradual.

It is critical that the change in tempering parameter $\Delta\beta_j$ is smooth between iterations such that the target posterior distribution does not change dramatically. Ching and Chen [7] proposed that the optimal $\Delta\beta_j$ is found such that the coefficient of variance of likelihood evaluations is 100%:

$$COV(\beta_j) = \frac{\sigma[P(D|\theta_i)^{\Delta\beta_j}]}{\mu[P(D|\theta_i)^{\Delta\beta_j}]} \tag{17}$$

$$1 = \frac{\sigma[P(D|\theta_i)^{\Delta\beta_j}]}{\mu[P(D|\theta_i)^{\Delta\beta_j}]} \tag{18}$$

$\Delta\beta_j$ can hence be found by solving the following equation:

$$f(\Delta\beta_j) = \sigma[P(D|\theta_i)^{\Delta\beta_j}] - \mu[P(D|\theta_i)^{\Delta\beta_j}] \tag{19}$$

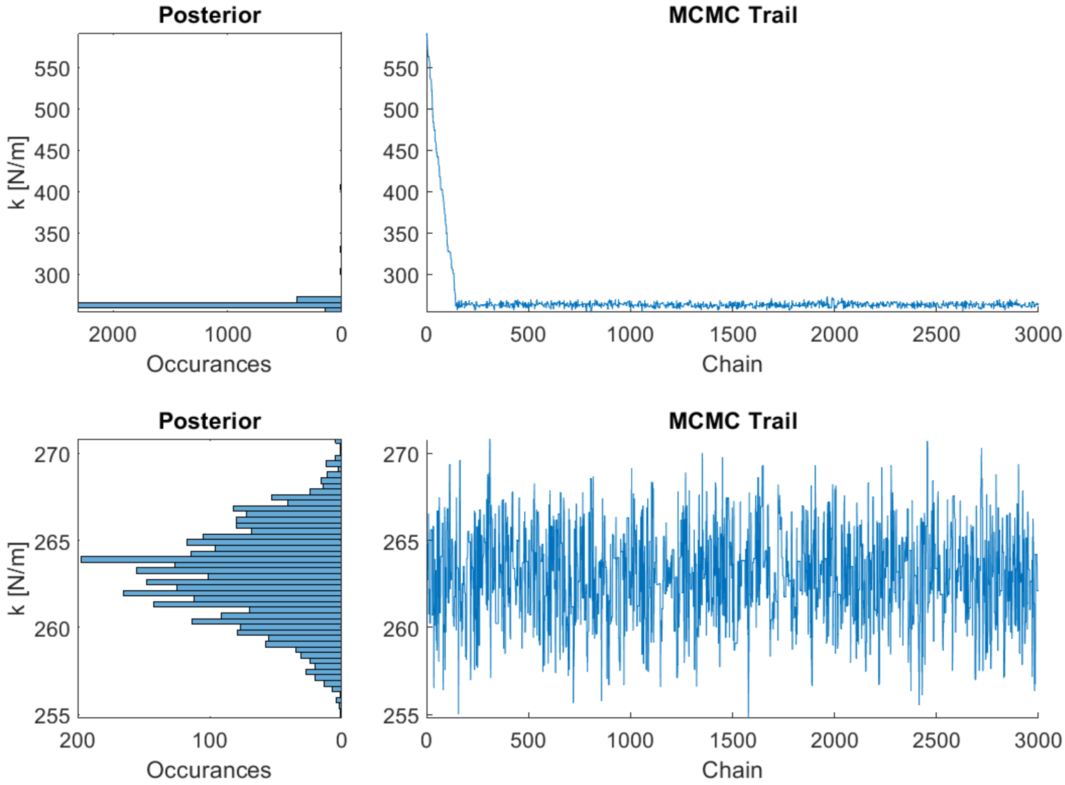

Figure 1: MCMC sampling to fit a simple linear-spring model ($F = -kx$), where k is the spring stiffness [Nm] and x is the displacement of the spring [m]. The spring has a true stiffness of 263 N/m. An initial, far-out guess of 600 N/m is taken to effectively demonstrate burn-in. **(Top)** Histogram and trail of MCMC algorithm with 0 burn-in. Note the long tail at the start of the trail which is a result of the MH algorithm gradually converging the proposal PDF to within the posterior bounds. **(Bottom)** Histogram and trail of MCMC algorithm with 200 sample burn-in, meaning, the first 200 samples are discarded from the final dataset.

In TMCMC's very first iteration, samples are taken from the prior before any transitions have taken place in order to evaluate the 'landscape' of the posterior. With this, importance weighting can then take place to begin establishing points of the posterior that hold significance, ensuring importance is given to these areas of the parameter space which show highly probable likelihoods when conducting re-sampling (Figure 6). Importance sampling weights are simply calculated by their likelihood value normalized against the sum of likelihood values for all samples:

$$\hat{w}(\theta_i) = \frac{P(D|\theta_i)^{\Delta\beta_j}}{\sum_{i=1}^{N} P(D|\theta_i)^{\Delta\beta_j}} \tag{20}$$

An N number of samples are re-sampled according to the calculated weightings. For each re-sampled parameter θ_i, a single-step Markov chain is run, with the sampled parameter used as the starting point. The logic behind running multiple, small Markov chains is to overcome problems encountered during typical MCMC algorithms: correlation of samples and chains getting 'stuck' within peaks. By initiating chains across the entire posterior instead of beginning in one single area, each chain can explore the several peaks which may potentially exist in the posterior. The proposal distribution for MCMC chains is a Normal distribution with a mean centered around the current sample and covariance matrix defined as:

$$\Sigma = \gamma^2 \sum_{i=1}^{N} \hat{w}(\theta_i) \cdot [\{\theta_i - \overline{\theta}\} \times \{\theta_i - \overline{\theta}\}^T] \tag{21}$$

where γ is a scaling parameter, the optimal value of which is 0.2 [7], and $\overline{\theta}$ is the mean of the sample set and is calculated by:

$$\overline{\theta} = \sum_{i=1}^{N} \theta_i \cdot \hat{w}(\theta_i) \tag{22}$$

2.4 Improved Transitional Markov-chain Monte Carlo (iTMCMC)

Betz, Papaioannou and Straub [8] proposed three modifications to the TMCMC algorithm which reduce the bias from samples: samples weights adjusted after each MCMC step, a burn-in period for MCMC sampling, and an adaptive scale for the proposal distribution. According to examples presented in the proposal paper, such changes improve the convergence behaviour of posterior samples while also achieving near-optimal acceptance rates and thus improved sampling efficiency. The merit of these modifications has been discussed by both sets of authors for TMCMC and iTMCMC, with a general agreement being that the scale of improvements are bound by the updating case at hand. Henceforth, this study will evaluate if such modifications are noteworthy in structural Bayesian model updating.

2.5 Bayesian Updating with Structural Methods (BUS)

Bayesian Updating with Structural Methods (BUS) translates Bayesian model updating problems into the realm of structural reliability analysis, carrying the advantage of allowing the use of existing and well-established structural reliability tools [9]. One such tool is subset simulation [21] which allows for efficient sampling in large, high dimensional parameter spaces with relatively small domains of interest.

Letting P denote a uniformly random variable in the interval [0, 1], the domain in which samples are accepted or rejected similarly to the Metropolis-Hastings algorithm is defined as:

$$\Omega = [P \leq cL(\mathbf{X})] \tag{23}$$

where c is a constant which ensures that $cL(\mathbf{x})$ is always less than or equal to 1. Generally, an optimal choice of c is inversely proportional to the largest possible value of the likelihood ($c = 1/\sup L(x)$). The ease of evaluating $\sup L(x)$ depends on a case by case basis and in the case present here it isn't readily available. For the results in this work, $\sup L(x)$ was taken as the largest likelihood value found from the samples produced in the current iteration.

BUS transforms the input and output space to a space of independent standard normal random variables ($\mathbf{U} = [U_0; u_1; ...U_n] \in \mathbb{R}^{n+1}$) to assist with the integration with structural reliability methods. To transform from $\mathbf{U}$ to P we apply the standard normal cumulative density function (CDF):

$$P = \Phi(U_0) \tag{24}$$

and to transform from $\mathbf{U}$ to $\mathbf{X}$:

$$\mathbf{X} = \mathbf{T}(U_1, ..., U_n) \tag{25}$$

where $\mathbf{T}$ is a transformation such as the Nataf transformation [22]. These transformations thus define the domain of interest as:

$$\Omega_U = \{\Phi(u_0) \leq cL\left[\mathbf{T}(u_1, ..., u_n)\right]\} \tag{26}$$

The domain Ω_U can be defined by a limit state function to aid integration into the model updating framework:

$$H(\mathbf{u}) = u_0 - \Phi^{-1}\{cL\left[\mathbf{T}(u_1, ..., u_n)\right]\} \tag{27}$$

such that sample $\mathbf{u}$ is in domain Ω_U when $[H(\mathbf{u}) \leq 0]$.

Subset simulation breaks the generally relatively small probability of a rare event Z_e into the product of many, larger probabilities which are easier to simulate.

$$Pr(Z_e) = Pr\left(\bigcap_{i=1}^{M} Z_i\right) = \prod_{i=1}^{M} Pr(Z_i|Z_{i-1}) \tag{28}$$

Within each subset, there is a conditional distribution from which samples are required. Applying this to the general BUS method, conditional domains are introduced such that a sample $\mathbf{u}$ is in domain Ω_i when $[H(\mathbf{u}) \leq h_i]$ where h_i is the current threshold level. h_i is set to the p_0 percentile of sorted samples from the current level, and is evaluated iteratively until $h_i = 0$. Very small values of p_0 leads to fewer iterations but more samples per iteration required, while large values has the opposite effect. Hence, a value which offers a suitable balance of iterations and samples is desired. The original BUS proposal paper found a suitable p_0 setting to be 0.1 for most cases, although values in the range of $p_0 \in [0.1, 0.3]$ achieve similar sampling efficiency [23].

Papaioannou et al. [24] proposed adaptive MCMC algorithms specifically for subset simulation which improve reduce bias and variance while improving efficiency and are hence optimized for subset simulation. Here, their proposed adaptive conditional sampling (aCS) is utilised to sample from conditional distributions. For specifics on this sampling method, readers are guided to their paper.

2.6 Adaptive Bayesian Updating with Structural Methods (aBUS)

Adaptive Bayesian Updating with Structural Methods (aBUS) attempts to optimally select a scaling constant, c, which is used to scale the likelihood function, for each threshold level within the updating process. A value of c which is too small negatively impacts the efficiency of sampling based BUS approaches and it is hence desirable to have an optimal value adaptively set throughout the updating process.

Similar to the standard BUS algorithm, the first step draws an N number of samples from user-provided prior distributions and the likelihood of these samples is evaluated and stored. To determine the first threshold level h_i, an initial value for c is calculated by considering the largest likelihood within the initial samples:

$$c_0 = \max(c, L(\theta_0|d)) \tag{29}$$

with h_i then calculated as the p_0 percentile of the likelihood evaluations.

In the original BUS implementation, this value of c does not change after it has been set. With aBUS, it is re-evaluated similarly after sampling has occurred on the current threshold level at iteration i:

$$c_{i+1} = \max(c, L(\theta_i|d)) \tag{30}$$

meaning that the threshold level for the next corresponding iteration is then determined by:

$$h_{i+1} = h_i - c_i + c_{i+1} \tag{31}$$

3 RESULTS

All results were generated on a computer with an Intel Core i7-12700 12-core processor with 24 total threads. TMCMC updating was performed with parallelism similar to Goller et al. [25], while the nature of the other sampling algorithms do not allow for parallelism.

3.1 Case 1: 3 degree-of-freedom system

The first test case is a simple 3 degree of freedom (DOF) spring-mass system as shown in Figure 2. Stiffness parameters k_1, k_2 and k_3 are considered as uncertain and updatable parameters while the mass parameters and remaining stiffness parameters are kept constant: $m_1 = 0.7$ kg, $m_2 = 0.5$ kg, $m_3 = 0.3$ kg and $k_{4-6} = 5.0$ Nm^{-1}. All three natural frequencies f_1, f_2 and f_3 are taken as outputs of interest. The uncertainty in k_{1-3} is assumed to each be of a Gaussian distribution with unknown means and variances which will be corrected by the model updating framework within intervals $\mu_{1-3} \in [3.0, 7.0]$ and $\sigma_{1-3} \in [0.0, 0.5]$. To facilitate updating, an 'experimental' dataset of 100 reference outputs is generated with the distribution properties of the uncertain parameters set to known values of $\mu_{1-3} = [4.0, 5.0, 6.0]$ and $\sigma_{1-3} = [0.3, 0.1, 0.2]$. When updating, each sampled parameter set is used to generate 100 model outputs for comparison with the experimental data.

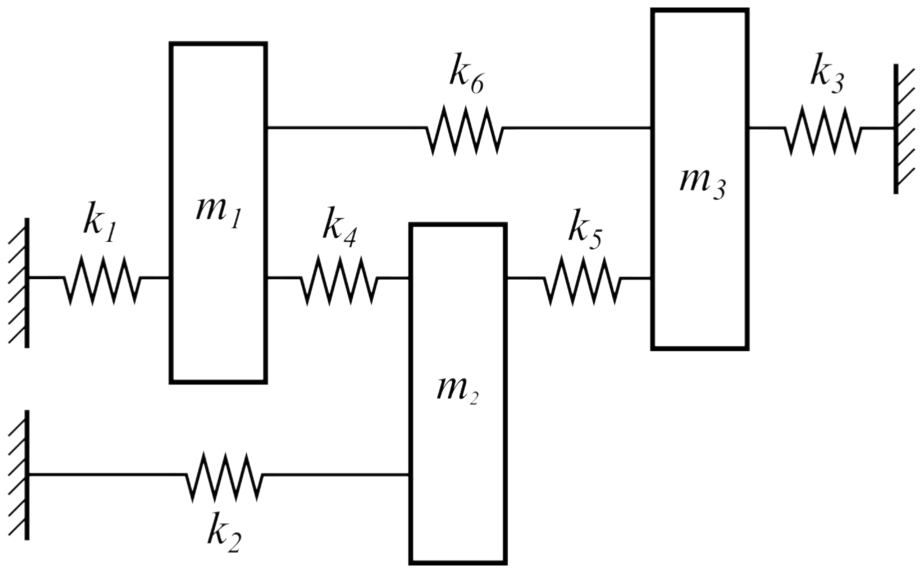

Figure 2: 3 degree-of-freedom mass and spring system.

MCMC sampling was performed with a burn-in period of 100, and a Normal proposal distribution with standard deviation equal to 1/50th of the parameter's interval. A total of 10,000

	Time [s]	Samples per iteration	Total samples	$L(x)$ evaluations
MCMC	7.49	10,000	20,000	40,200
TMCMC	176.87	1,000	34,000	102,000
iTMCMC	807.88	1,000	35,000	455,000
BUS	1590.32	10,000	130,000	270,000
aBUS	1734.61	10,000	135,000	280,000

Table 1: Computational cost metrics for all sampling methods when updating the 3 degree of freedom spring-mass system. 'Iterations' is a combination of the total iterations required in both Euclidean distance updating and Bhattacharyya updating steps. Note that for likelihood function evaluation the core numerical model is ran several times, so the number of model evaluations is much higher.

samples were obtained for the Euclidean and Bhatacharyya steps individually. Posterior distributions for mean parameters clearly converge around the true value and resemble normal distributions, while posterior samples for variance parameters struggle to converge around true values. Despite the relatively large burn-in period, samples for variance parameters still exhibit 'walking' characteristics while samples slowly converge to a minima.

TMCMC sampling was performed with 1,000 samples assigned to each updating step. All updated parameter's posteriors show clean convergence to a Normal-like distribution shape with peaks in good agreement with true values.

Similar to TMCMC, iTMCMC sampling was performed with 1,000 samples in each updating step to allow for direct comparison of the proposed improvements. A burn-in period of 10 samples was utilised. As iTMCMC runs a Markov chain for each sample in an iteration, this resulted in an extra $N_b \times N_s$ samples, where N_b is the number of burn-in sample and N_s is the number of samples per iteration. Despite the now included burn-in period, variance parameters show long tails at the base of their posterior distributions as the samples converge around the true values. Furthermore, the width of said posterior bases appears wider when compared to TMCMC results, suggesting larger variance in samples.

Initial runs of BUS sampling with default hyperparameters (1,000 samples, conditional probability $p_0 = 0.2$ noted low acceptance rates in MCMC threshold sampling and thus poor sampling results with sparsely filled distributions. Because of this, iterations required more samples be to create acceptable posterior distribution shapes. A setting of 10,000 samples per iteration was found to be the minimum where posteriors where properly developed. The final results in Figure 3 show general good agreement with true parameter values, but with a large spread in variance parameters.

Similarly to BUS, aBUS sampling required modified conditional probabilities and iteration sample sizes, with these being set to the same as BUS. Compared to standard BUS, distributions are better formed and better centered near true values, although not as precisely as TMCMC or iTMCMC.

Model outputs generated with the resultant updated parameters (Figure 4) show result clouds which are generally all in agreement with the generated experimental samples. MCMC's failure to closely calibrate the variance parameters shows with its results distribution orientated at a slightly different angle and slightly wider variance with outliers outwith the experimental cloud.

3.2 Case 2: Stochastic model updating benchmark testbed

To enable comparison of each sampling method's relative performance, model updating is performed upon a small-scale aeroplane model, designed to be a new benchmark stochastic

	Parameter	True Value	Max-p value	Error	Avg. Abs. Error
MCMC	k_1^μ	4.0	3.9684	0.765%	17.84%
	k_2^μ	5.0	4.9943	-0.404%	
	k_3^μ	6.0	6.0074	-0.245%	
	k_1^σ	0.3	0.2841	17.9%	
	k_2^σ	0.1	0.3526	66.1%	
	k_3^σ	0.2	0.3768	-21.65%	
TMCMC	k_1^μ	4.0	4.0189	0.473%	6.13%
	k_2^μ	5.0	4.9800	-0.4%	
	k_3^μ	6.0	6.0097	0.162%	
	k_1^σ	0.3	0.2794	-6.867%	
	k_2^σ	0.1	0.0860	-14%	
	k_3^σ	0.2	0.2297	14.85%	
iTMCMC	k_1^μ	4.0	3.9680	-0.8%	7.12%
	k_2^μ	5.0	5.0138	0.276%	
	k_3^μ	6.0	5.9751	-0.415%	
	k_1^σ	0.3	0.3321	10.7%	
	k_2^σ	0.1	0.1208	20.8%	
	k_3^σ	0.2	0.2195	9.75%	
BUS	k_1^μ	4.0	3.9784	-0.54%	7.80%
	k_2^μ	5.0	5.0071	0.142%	
	k_3^μ	6.0	5.9770	-0.383%	
	k_1^σ	0.3	0.3819	27.3%	
	k_2^σ	0.1	0.0856	-14.4%	
	k_3^σ	0.2	0.1919	-4.05%	
aBUS	k_1^μ	4.0	3.9829	-0.428%	6.78%
	k_2^μ	5.0	5.0152	0.304%	
	k_3^μ	6.0	6.0459	0.765%	
	k_1^σ	0.3	0.2932	-2.267%	
	k_2^σ	0.1	0.1277	27.7%	
	k_3^σ	0.2	0.1816	-9.2%	

Table 2: Updated parameters for all sampling methods when updating the 3 degree of freedom spring-mass test case.

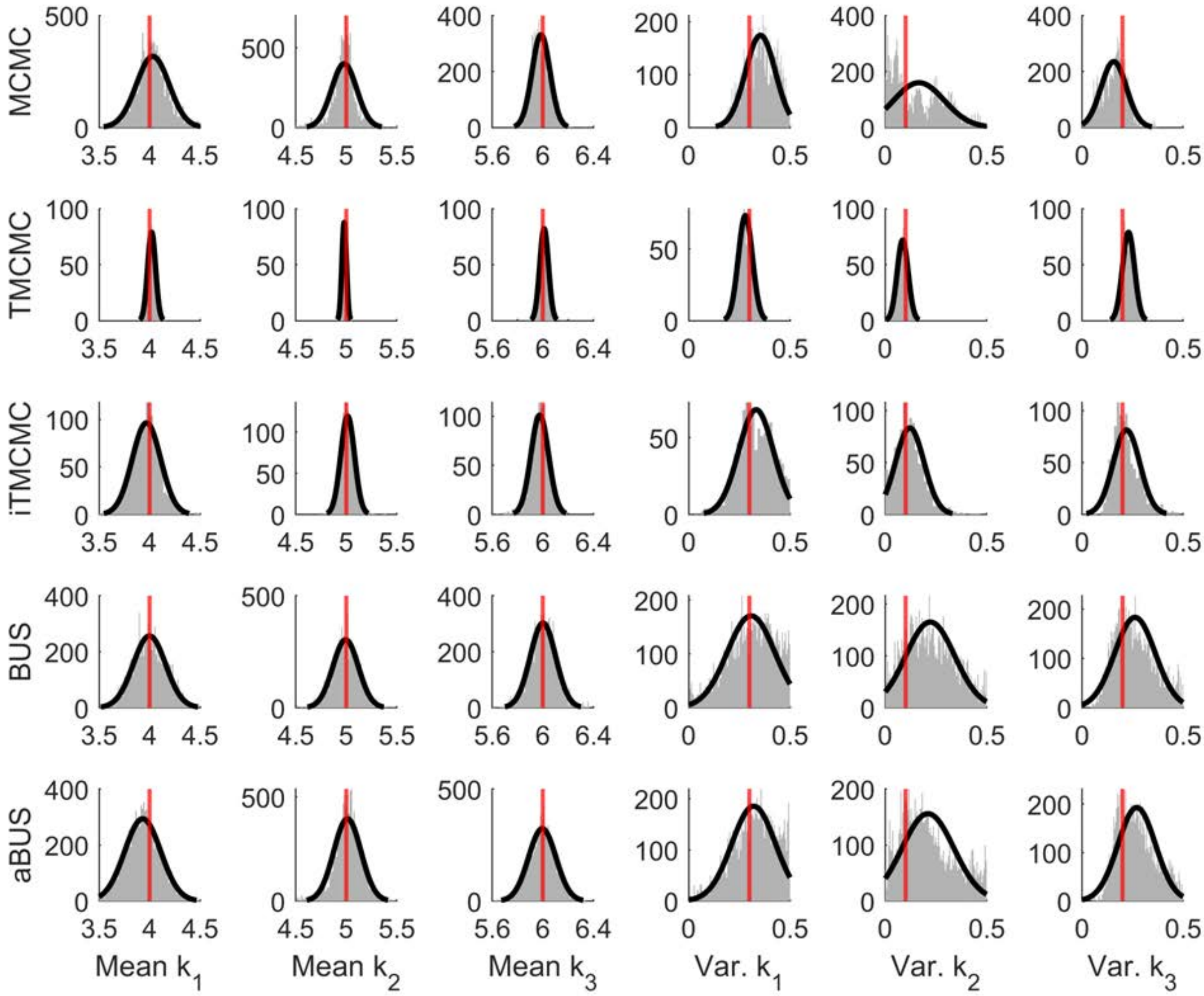

Figure 3: Posterior distributions for the means and variances of input parameters k_1, k_2 and k_3 after updating the 3 degree of freedom spring-mass model. Red line indicates parameter's target (true) value.

model within the community. Despite the wealth of research offered into the stochastic model updating community, the new insights offered are typically only compared against pure numerical examples with 'fake', generated experimental results used to update models rather than physical data with truly unknown parameters. Such lack of physical benchmarks is extremely noticeable in the aerospace context, with the GARTEUR test structure [26] from 1997 being perhaps the only benchmark available which is based upon physical measurements. As such, we seek to introduce a new benchmark to the stochastic model updating community with which novel sampling methods and uncertainty quantification measures can be developed and compared. The true values used to generate the experimental outputs will be unpublished to avoid confirmation bias when generating solutions.

Rather than a single deterministic structure, the model has 30 wing configurations with varying geometry properties following predefined probabilistic distributions. Since the wings of different shapes were disassembled and reassembled for each modal test, the measurement dataset includes not only "controllable" uncertainty originating from the structural property itself, but also experimental uncertainty. The model updating task of this testbed is stochastic by nature -

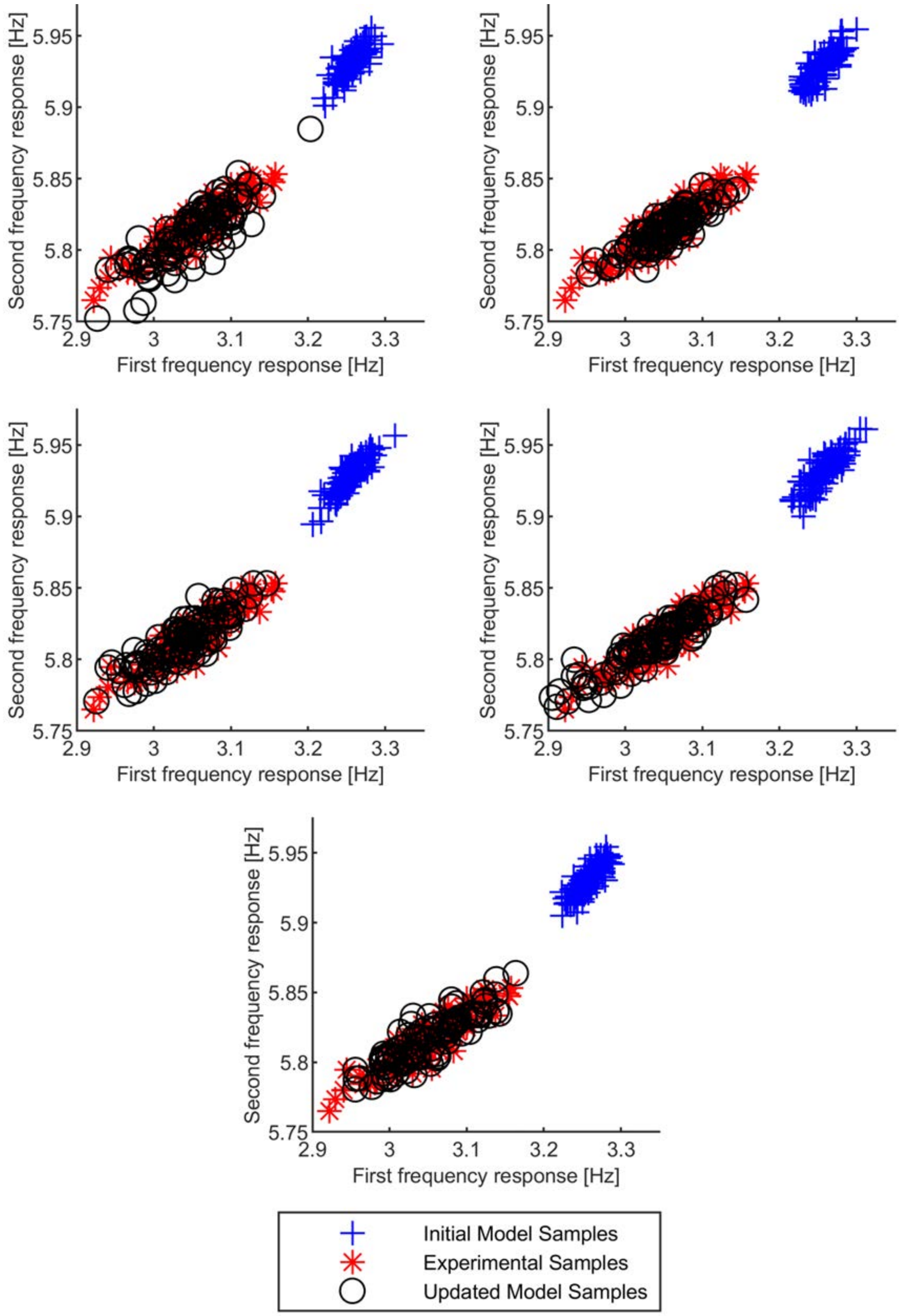

Figure 4: 3 degree-of-freedom model first and secondary frequency responses for initial iteration and final updated parameters.

to find the true designed distribution properties with aleatoric uncertainties in the wingspan and width of wing, as well as further uncertainties in FE modelling simplifications, experimental randomness, data acquisition, and signal processing.

The numerical model of the aircraft was defined as a finite element (FE) model within ANSYS Mechanical 2022 R2. Due to the majority thin thickness of components, a surface based model and shell elements were preferred. A modal analysis was setup in a 'free-free' configuration wherein no boundary conditions are applied - post-computation the first six natural frequencies are discarded as these correspond to translation and rotation in the X, Y, and Z directions/axes respectively and are not experimentally measured frequencies. A mesh convergence study took place to find a mesh resolution which provided a suitable balance between fast simulation runs and model fidelity. At connection points between the winglets to the fuselage and between the wing and fuselage, a secondary material was set. The motivation behind this second material is to open this material's Young's Modulus as an updated parameter which can hopefully correct for the unavoidable uncertainties introduced by modelling assumptions and simplifications. Wingspan and wing width dimensions, as well as the Young's Modulus for the secondary material were exposed as external parameters for parametric study via a linking with MATLAB R2022b.

Due to the relatively large computational effort required when identifying modal frequencies via the full-fidelity numerical model, a surrogate model is preferred, especially when sampling is likely to take several tens of thousands of evaluations. Goller *et. al* [25] suggest that feed-forward neural networks, where each model output is given its own surrogate model, are an accurate method for predicting model response. Using the geometric and material parameters as inputs, an automated optimisation strategy was utilised to optimise network topology and hyperparameters while minimising cross-validation mean squared error. This strategy is ran across 1000 samples obtained from the numerical model, with 200 reserved for model validation.

For model updating, the uncertainty in wingspan and wing width is known to follow a Multivariate Normal distribution with correlation between the parameters. With this in mind, a total of 6 parameters are opened for updating: the mean wingspan $a_\mu \in [280, 320]$, the variance in wingspan $a_\sigma \in [0, 5]$, the mean wing width $b_\mu \in [15, 35]$, the variance in wing width $b_\sigma \in [0, 5]$, the correlation $\rho \in [-1, 1]$ and the Young's Modulus of connections $E \in [10 \times 10^9, 120 \times 10^9]$. When updating, each sampled parameter set is used to generate 30 model outputs for comparison with the experimental data.

The updated models (Figure 5) obtained by all sampling methods were able to create stochastic outputs which where within the same range as the experimentally measured outputs. MCMC updating - utilising 10,000 samples per distance updating step, a burn-in period of 100 samples and Normal proposal distributions with a standard deviation equal to 1/10th of parameter intervals - created outputs with means that were close and variances which almost exactly match the measured frequency response. The updated model obtained by TMCMC - with 1,000 samples per iteration - showcased similar characteristics, albeit while taking over double the time (Table 3). iTMCMC, updating output means closely to experimental measurements, struggled to correctly calibrate the variance in outputs. However, the extra computational effort involved in constantly recalculating weights and proposal scales resulted in a dramatic increase in runtime. Similarly to the simpler spring-mass system test case, BUS and aBUS struggled to sufficiently sample posteriors when using similar sample sizes as TMCMC, requiring a larger amount of samples per iteration. The more complex case study resulted in BUS struggling to calibrate both means and variances, and the adaptivity brought by aBUS completely deteriorated variance calibration and no output means tightly matched.

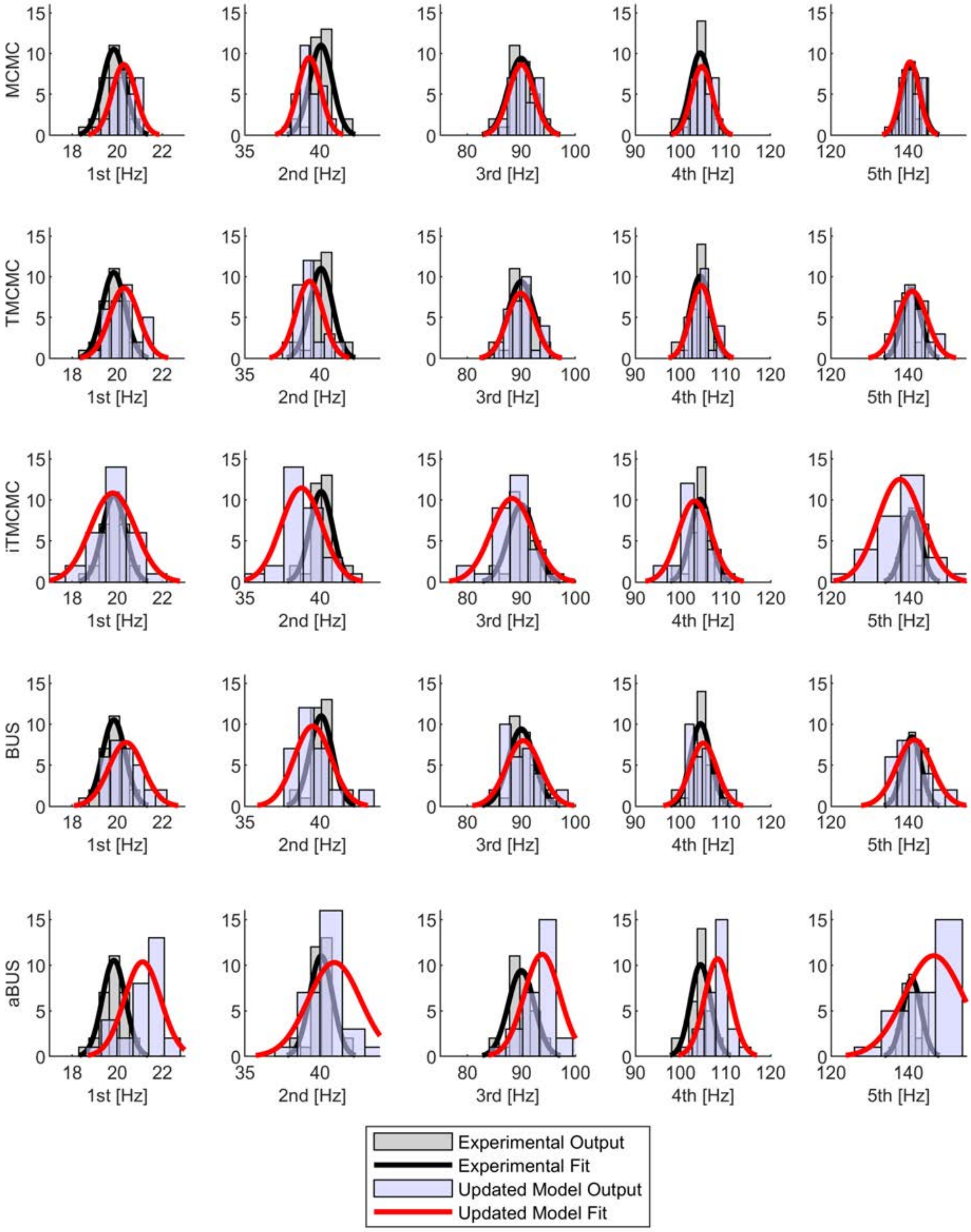

Figure 5: Experimentally measured and updated model frequency response histograms for airplane test case.

	Time [s]	Samples per iteration	Total samples	$L(x)$ evaluations
MCMC	110.767	10,000	20,000	40,200
TMCMC	406.4	1,000	32,000	96,000
iTMCMC	9643.674	1,000	58,000	174,000
BUS	690.411	10,000	140,000	160,000
aBUS	576.79	10,000	120,000	140,000

Table 3: Computational cost metrics for all sampling methods when updating the airplane model. 'Iterations' is a combination of the total iterations required in both Euclidean distance updating and Bhattacharyya updating steps. Note that for likelihood function evaluation the core numerical model is ran several times, so the number of model evaluations is much higher.

4 DISCUSSION

All five sampling methods proved effective at updating the three degree of freedom spring-mass test case to match generated experimental outputs, even if the exact model parameters used to generate experimental outputs weren't always met. TMCMC produced what are arguably the best set of results which feature clear and smooth distributions with tight spreads, simultaneously providing modes which matched true values the closest (Table 2). As a result, it comes as no surprise that TMCMC's updated model outputs tightly fit the measured experimental response. That being said, the objective of model updating isn't necessarily to match unknown input parameters but to match the measured outputs. MCMC's posterior results, while not as neatly distributed as TMCMC nor as close to the true values, still show reasonable correlation with the experimental response at a fraction of the computational cost (Table 1). The largest drawback to MCMC in this test case was the method's sensitivity to well set proposal distribution. While using Normal distributions with standard deviations equal to 1/50th of parameters' intervals was found to work reasonably well, the process to reach these values took considerable trial and error. This scenario could have improved with an adaptive proposal distribution [27], however the investigation of these was outwith the scope of this study. Indeed, an advantage of the other presented approaches is that they are effectively 'plug-and-play' and don't require fine tuning of proposal distributions.

Despite the inclusion of a burn-in for the iTMCMC method, a clear tail of converging samples could be observed when updating the three degree of freedom test case. It is possible that a burn-in period larger than 10 may have reduced this artifact upon the final posterior, albeit with the drawback of significant computational cost. The proposed improvements had a detrimental effect upon the calibration of the airplane model, suggesting that the improvements aren't necessarily suitable for every usage case. Indeed, this is something noted by the original authors in a discussion to the improvements paper [28].

Initial runs of BUS and aBUS updating on both test cases noted low Metropolis-Hastings acceptance rates (around 5-8%) during the final iterations of sampling - far from the generally optimal rate of 44%. Because of this, the number of samples had to be increased to sizes larger than what TMCMC required, increasing computational cost in comparison. With this, it is hard to see the benefit of the additional complexity brought on with BUS when compared to MCMC and TMCMC in the context of this test case. However, because BUS extends structural methods into the Bayesian updating world, subset simulation is not the only reliability analysis tool that can be utilised and it may be interesting to evaluate the performance of BUS with other developments such as line sampling [29]. Indeed, it appears that subset simulation struggles to sample efficiently and effectively with the Bhattacharyya distance based likelihood, with further investigation required to determine why this may be and whether alternative reliability analysis methods could have more success in doing so. aBUS was originally proposed to improve efficiency due to the potential for fewer threshold levels - although this was only true for the airplane model test case where 2 fewer levels were required, while for the 3 degree of freedom spring-mass system 2 more levels were required.

At first glance, one could conclude that, at least for the presented cases, basic MCMC calibrates models to within an acceptable level of accuracy while being by far the most computationally efficient method. However, it was noted that the updated results varied greatly with every run of the MCMC updating program. A useful extension of the given investigation would be a quantitative analysis of the robustness of each method to ensure suitable model updating results are repeatable.

Giving consideration to the model updating framework used, the Euclidean distance remains a solid metric for the calibration of parameter means, with mean inputs closely matching true values in the spring-mass test case, and output means also close matching in the airplane test case. The Bhattacharyya distance, while successfully capturing the variance in the spring-mass system case, struggled to fully calibrate the variance in all of the airplane's model outputs. This is potentially due to the lower quantity of experimental measurements (30 compared to 100). As such, it would be fruitful for future developments in Bayesian model updating to push towards correctly capturing variance in sparse datasets, a situation common in aerospace where data is prohibitively expensive to collect.

5 CONCLUSIONS

In this study the performance of several sampling methods was compared for use within a Bayesian model updating framework. All methods were able to successfully calibrate the model of a simple test case involving a three degree of freedom spring-mass system to an acceptable degree of accuracy. MCMC proved to do so with the least computational effort, although being relatively the most inaccurate. TMCMC remains a robust and easy to implement sampling method with few hyperparameters requiring tuning. The improvements brought by iTMCMC are dependent on the case study at hand and any improvement in accuracy is reduced by the increased computational cost. The ability to easily parallelise TMCMC's operation makes it a straight-forward choice when the hardware to do so is available. BUS and its adaptive variant aBUS offer an alternative take on the Bayesian updating problem by utilising structural reliability methods, however it proved inefficient for the case at hand. Different implementations other than the present subset simulation method offer interesting further research.

The difficulty in successfully updating uncertainty parameters in the test aeroplane model suggests that the state-of-the-art in stochastic model updating still needs refinement in such test cases. While the Euclidean distance updates parameter means with ease, the Bhattacharyya distance is not always successful in updating parameter variances, especially with limited datasets such as is presented here. Alternative measurements are desired which make full use of the limited dataset and the development of such statistics is left as future work for the authors and members of the community.

REFERENCES

[1] L. Le Cam. Maximum Likelihood: An Introduction. *International Statistical Review / Revue Internationale de Statistique*, 58(2):153–171, 1990. Publisher: [Wiley, International Statistical Institute (ISI)].

[2] Javier Fernando Jiménez-Alonso, Javier Naranjo-Perez, Aleksandar Pavic, and Andrés Sáez. Maximum Likelihood Finite-Element Model Updating of Civil Engineering Structures Using Nature-Inspired Computational Algorithms. *Structural Engineering International*, 31(3):326–338, July 2021. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10168664.2020.1768812.

[3] J. E. Mottershead and M. I. Friswell. Model Updating In Structural Dynamics: A Survey. *Journal of Sound and Vibration*, 167(2):347–375, October 1993.

[4] John E. Mottershead, Michael Link, and Michael I. Friswell. The sensitivity method in finite element model updating: A tutorial. *Mechanical Systems and Signal Processing*, 25(7):2275–2296, October 2011.

[5] Sifeng Bi and Michael Beer. Overview of Stochastic Model Updating in Aerospace Application Under Uncertainty Treatment. In Louis J. M. Aslett, Frank P. A. Coolen, and Jasper De Bock, editors, *Uncertainty in Engineering: Introduction to Methods and Applications*, SpringerBriefs in Statistics, pages 115–129. Springer International Publishing, Cham, 2022.

[6] Christian Robert and George Casella. A Short History of Markov Chain Monte Carlo: Subjective Recollections from Incomplete Data. *Statistical Science*, 26(1):102–115, 2011. Publisher: Institute of Mathematical Statistics.

[7] Jianye Ching and Yi-Chu Chen. Transitional Markov Chain Monte Carlo Method for Bayesian Model Updating, Model Class Selection, and Model Averaging. *Journal of Engineering Mechanics*, 133(7):816–832, July 2007. Publisher: American Society of Civil Engineers.

[8] Wolfgang Betz, Iason Papaioannou, and Daniel Straub. Transitional Markov Chain Monte Carlo: Observations and Improvements. *Journal of Engineering Mechanics*, 142(5):04016016, May 2016. Publisher: American Society of Civil Engineers.

[9] Daniel Straub and Iason Papaioannou. Bayesian Updating with Structural Reliability Methods. *Journal of Engineering Mechanics*, 141(3):04014134, March 2015. Publisher: American Society of Civil Engineers.

[10] Wolfgang Betz, Iason Papaioannou, James L. Beck, and Daniel Straub. Bayesian inference with Subset Simulation: Strategies and improvements. *Computer Methods in Applied Mechanics and Engineering*, 331:72–93, April 2018.

[11] Francesca Bianconi, Georgios Panagiotis Salachoris, Francesco Clementi, and Stefano Lenci. A Genetic Algorithm Procedure for the Automatic Updating of FEM Based on Ambient Vibration Tests. *Sensors*, 20(11):3315, January 2020. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

[12] Hakan Gökdağ and Ali R. Yildiz. Structural Damage Detection Using Modal Parameters and Particle Swarm Optimization. *Materials Testing*, 54(6):416–420, June 2012. Publisher: De Gruyter.

[13] R. I. Levin and N. A. J. Lieven. DYNAMIC FINITE ELEMENT MODEL UPDATING USING SIMULATED ANNEALING AND GENETIC ALGORITHMS. *Mechanical Systems and Signal Processing*, 12(1):91–120, January 1998.

[14] Javier Naranjo-Pérez, Javier F. Jiménez-Alonso, Aleksandar Pavic, and Andrés Sáez. Finite-element-model updating of civil engineering structures using a hybrid UKF-HS algorithm. *Structure and Infrastructure Engineering*, 17(5):620–637, May 2021. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/15732479.2020.1760317.

[15] Adolphus Lye, Alice Cicirello, and Edoardo Patelli. Sampling methods for solving Bayesian model updating problems: A tutorial. *Mechanical Systems and Signal Processing*, 159:107760, October 2021.

[16] Cosmin Safta, Khachik Sargsyan, Habib N. Najm, Kenny Chowdhary, Bert Debusschere, Laura P. Swiler, and Michael S. Eldred. Probabilistic Methods for Sensitivity Analysis and Calibration in the NASA Challenge Problem. *Journal of Aerospace Information Systems*, 12(1):219–234, January 2015. Publisher: American Institute of Aeronautics and Astronautics.

[17] Inseok Park and Ramana V. Grandhi. Quantifying Multiple Types of Uncertainty in Physics-Based Simulation Using Bayesian Model Averaging. *AIAA Journal*, 49(5):1038–1045, May 2011. Publisher: American Institute of Aeronautics and Astronautics.

[18] A. Bhattacharyya. On a Measure of Divergence between Two Multinomial Populations. *Sankhyā: The Indian Journal of Statistics (1933-1960)*, 7(4):401–406, 1946. Publisher: Springer.

[19] Sifeng Bi, Matteo Broggi, and Michael Beer. The role of the Bhattacharyya distance in stochastic model updating. *Mechanical Systems and Signal Processing*, 117:437–452, February 2019.

[20] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092, June 1953. Publisher: American Institute of Physics.

[21] Siu-Kui Au and James L. Beck. Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics*, 16(4):263–277, October 2001.

[22] Ove Ditlevsen and Henrik O. Madsen. *Structural Reliability Methods*. January 1996.

[23] Konstantin M. Zuev, James L. Beck, Siu-Kui Au, and Lambros S. Katafygiotis. Bayesian post-processor and other enhancements of Subset Simulation for estimating failure probabilities in high dimensions. *Computers & Structures*, 92-93:283–296, February 2012.

[24] Iason Papaioannou, Wolfgang Betz, Kilian Zwirglmaier, and Daniel Straub. MCMC algorithms for Subset Simulation. *Probabilistic Engineering Mechanics*, 41:89–103, July 2015.

[25] B. Goller, M. Broggi, A. Calvi, and G. I. Schuëller. A stochastic model updating technique for complex aerospace structures. *Finite Elements in Analysis and Design*, 47(7):739–752, July 2011.

[26] Etienne Balmes. GARTEUR Group on Ground Vibration Testing. Results from the Test of a Single Structure by 12 Laboratories in Europe. January 1997.

[27] Christophe Andrieu and Johannes Thoms. A tutorial on adaptive MCMC. *Statistics and Computing*, 18(4):343–373, December 2008.

[28] Jianye Ching and Jiun-Shiang Wang. Discussion of "Transitional Markov Chain Monte Carlo: Observations and Improvements" by Wolfgang Betz, Iason Papaioannou, and Daniel Straub. *Journal of Engineering Mechanics*, 143(9):07017001, September 2017. Publisher: American Society of Civil Engineers.

[29] Marco de Angelis, Edoardo Patelli, and Michael Beer. Advanced Line Sampling for efficient robust reliability analysis. *Structural Safety*, 52:170–182, January 2015.