Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 199 (2017) 978–983

Procedia Engineering

www.elsevier.com/locate/procedia

X International Conference on Structural Dynamics, EURODYN 2017

Model Updating Strategy of the DLR-AIRMOD Test Structure

Edoardo Patelli[a,*], Matteo Broggi[b], Yves Govers[c], John E. Mottershead[a]

[a] *Institute for Risk and Uncertainty, University of Liverpool, Liverpool L69 7ZF, UK*
[b] *Institut für Risiko und Zuverlässigkeit, Leibniz University Hannover, Hannover 30167, Germany*
[c] *Institute of Aeroelasticity, German Aerospace Centre (DLR), Göttingen 37073, Germany*

Abstract

Considerable progresses have been made in computer-aided engineering for the high fidelity analysis of structures and systems. Traditionally, computer models are calibrated using deterministic procedures. However, different analysts produce different models based on different modelling approximations and assumptions. In addition, identically constructed structures and systems show different characteristic between each other. Hence, model updating needs to take account modelling and test-data variability.

Stochastic model updating techniques such as sensitivity approach and Bayesian updating are now recognised as powerful approaches able to deal with unavoidable uncertainty and variability. This paper presents a high fidelity surrogate model that allows to significantly reduce the computational costs associated with the Bayesian model updating technique. A set of Artificial Neural Networks are proposed to replace multi non-linear input-output relationships of finite element (FE) models. An application for updating the model parameters of the FE model of the DRL-AIRMOD structure is presented.

Keywords: Model updating; Artificial Neural Networks; Bayesian; Simulation

1. Introduction

The purpose of model updating is to adjust the computer model such that its outputs agree with experimental data obtained typically from a modal test. This is an essential step of model validation and verification. Today, different stochastic model updating procedures have been developed and applied in different fields (see e.g. [1–4]). The two most promising approaches to the stochastic model updating problem are the sensitivity (covariance updating) method [5] and Bayesian updating [6,7]. Recently, these two techniques have been combined in order to overcome the limitation of the individual procedure [1]. More specifically, the results of the sensitivity approach are used to define prior distributions for the Bayesian Model updating. By doing so, the computational efforts associated with the Bayesian Model updating procedure are largely reduced and the final distributions of the model parameters can evolve and relax the initial assumption of Gaussian parameter variability used in the sensitivity approach.

The main obstacle to practical engineering application of stochastic model updating is associated to the significant computer cost required. This is because many evaluations of a time-consuming model are required for performing un-

* Corresponding author. Tel.: +44 15179 44079.
E-mail address: edoardo.patelli@liverpool.ac.uk

1877-7058 © 2017 The Authors. Published by Elsevier Ltd.
Peer-review under responsibility of the organizing committee of EURODYN 2017.
10.1016/j.proeng.2017.09.221

certainty quantification. This is particularly true for Bayesian model updating, which until recently was restricted only to small-scale numerical examples. In fact, large scale applications of Bayesian Model updating requires advanced statistical sampling technique, parallel computing and high-fidelity surrogate models. This issue has been faced by replacing the full FE models with surrogate meta-models. Popular surrogate models use Artificial Neural Networks (ANN) [8,9], Polynomial Chaos [10], Kriging [11], etc and a number of applications in different fields to update large and complex models (see e.g. [12]) (see e.g. [2,12]) have already demonstrated the applicability of the stochastic model updating approach.

The focus of this paper is not to provide a comparison of different stochastic model updating strategies. Instead, the purpose is to present a technique for developing a high-fidelity surrogate model based on Artificial Neural Networks for the Bayesian model updating. This strategy has been successfully used in Ref. [1] for the parameter updating of the FE model of the DRL-AIRMOD structure [13].

1.1. Bayesian Model Updating

The Bayesian model updating is based on the well-known Bayes rule has been introduced by Beck and Katafygiotis [6,7]. The general formulation is the following:

$$P(\boldsymbol{\theta}|D,\mathcal{M}) = \frac{P(D|\boldsymbol{\theta},\mathcal{M})\,P(\boldsymbol{\theta}|\mathcal{M})}{P(D|\mathcal{M})} \tag{1}$$

where $\boldsymbol{\theta}$ represents any hypothesis to be tested, e.g. the value of the model parameters, D is the available experimental data (i.e. observations), and $\mathcal{M}$ is the chosen mathematical model. The different terms of Eq 1 are: the likelihood function of the data D ($P(D|\boldsymbol{\theta},\mathcal{M})$) representing the probability of obtaining the data D when the values of $\boldsymbol{\theta}$ are fixed for a chosen model $\mathcal{M}$; the prior probability density function of $\boldsymbol{\theta}$ ($P(\boldsymbol{\theta}|\mathcal{M})$ that incorporates prior information about the values of the parameters; the posterior probability density function of $\boldsymbol{\theta}$ ($P(\boldsymbol{\theta}|D,\mathcal{M})$) representing the new information based on observations and prior knowledge. Finally, the denominator term ($P(D|\mathcal{M})$) representing a normalization factor (also called evidence) ensuring that the posterior distribution function integrates to 1.

The equation (1) introduces a way to update some apriori knowledge on the parameters $\boldsymbol{\theta}$, by using data or observations D and conditional to some available information or hypothesis I. In the context of model updating, the initial knowledge about the unknown adjustable parameters, e.g. from prior expertise, is expressed through the prior PDF. A proper prior distribution can be a uniform distribution in the case when only a lower and upper bound of the parameter is known, or a Gaussian distribution when the mean and the relative error of the parameter are known. The data consists of the eigenvalue residuals, $\epsilon_i = \{z_i^e - z_i(\boldsymbol{\theta})\}$ where z_i^e typically represents the i-th measured natural frequency and $z_i(\boldsymbol{\theta})$ the i-th eigenvalue of the mathematical model.

The likelihood function gives a measure of the agreement between the available experimental data and the corresponding numerical model output. Particular care has to be taken in the definition of the likelihood, and the choice of likelihood depends on the type of data available, e.g. whether the data is a scalar or a vector quantity. Different likelihood leads to different accuracy and efficiency in the results of the updating procedure and should be selected with caution; as an example, the use of unsuitable likelihood function might cause that the model updating do not produce any relevant variation in the prior. The likelihood function is often chosen to be a multivariate normal distribution [6,12]. In practical application the data are also considered independent reducing the likelihood function to:

$$P(D|\boldsymbol{\theta},\mathcal{M}) = \prod_{i=1}^{n} \frac{1}{\sigma_i^2\sqrt{2\pi}} exp\left(-\frac{\epsilon_i^2}{2\sigma_i^2}\right) \tag{2}$$

Finally, the posterior distribution expresses the updated knowledge about the parameters, providing information on which parameter ranges are more probable based on the initial knowledge and the experimental data.

The solution of equation (1) for realistic structural dynamics problems is generally intractable analytically or numerically. Instead the solution is obtained by means of modern sampling techniques. Markov Chain Monte Carlo (MCMC) methods are the largely adopted procedures. The Metropolis-Hastings algorithm carries out a random walk in the model space sampled from the prior distribution, concentrated on regions of significant probability. Successive samples drawn from the so-called proposal distribution are dependent only upon the immediate past sample in the chain and independent of preceding samples (i.e. Markovian property). The algorithm picks a candidate next sample if the new sample is more probable (i.e. corresponding to large value of the posterior distribution) than the current sample. In the case of a rejection the sample remains unchanged and is used again in the next iteration. It can be shown that convergence is asymptotic independent of the proposal distribution used to generate test samples. Until quite recently, the application of Bayesian updating to industrial scale systems has not been possible because of the requirement for huge computing resource. This restriction has now been largely overcome by the development of modern stochastic simulation algorithm, called Transitional Markov Chain Monte Carlo (TMCMC) [14]. This algorithm allows the generation of samples from the complex shaped unknown posterior distribution through an iterative approach. In this algorithm, m intermediate distributions P_i are introduced:

$$P_i \propto P(D|\boldsymbol{\theta}, I)^{\beta_i} P(\boldsymbol{\theta}|I) \tag{3}$$

where the contribution of the likelihood is scaled down by an exponent β_i, with $0 = \beta_0 < .. < \beta_i < .. < \beta_m = 1$, thus the first distribution is the prior PDF, and the last is the posterior. The value of these exponents β_i is automatically selected to ensure that the dispersion of the samples at each step meet a prescribed target. By employing intermediate distributions, it is easier for the updating procedure to generate samples also from posterior showing very complex distribution, e.g., very peaked or multi-modal. For additional information the reader is reminded to [14].

1.2. Surrogate Model

Bayesian model updating procedures based on stochastic sampling require a huge number of model evaluations. For this reason surrogate models, also known in literature as meta-models or emulators, are widely used to mimic the input-output relation of computational expensive numerical model $\mathcal{M}$ by using a cheaper analytical model $\hat{\mathcal{M}}$. Some examples of surrogate models are Artificial Neural Networks (ANN), Poly-Harmonic Splines or Support Vector Machines. In this work, ANNs have been selected as surrogate model for their effectiveness and versatility.

ANN defines a function $f : X \rightarrow Y$, where $f(x)$ is a composition of other weighted functions $g(x)$. Often the non-linear weighted sum is used for the composition as follows:

$$\hat{y} = f(x) = K\left(\sum_i w_i \cdot g_i(x)\right) \tag{4}$$

where K is an activation function such as logistic or hyperbolic functions, $\hat{y}$ is the predicted output of the surrogate model w_i are weights to be updated in the ANN training algorithm. The classical architecture type consists of one input layer, one or more hidden layers and one output layer. Each layer employs several nodes, which are connected to the nodes of the adjacent layers by weighted links. In each node, the inputs $g_i(x)$ are first weighted and summed; then, the sum is processed by an activation function K to produce the node output. A bias is generally introduced in the hidden and output layers, which acts as threshold for the argument of activation function. The employed activation function is the widely used sigmoidal function $f(x) = (1 + e^{-x})^{-1}$. The selection of an ANN topology is general based on fitting performance. A simple approach consist of an heuristic testing of different topologies exploring some of the possible topological combinations. Then, the best ANN architecture is selected based on a performance indicator [9].

In the context of model updating, a complex numerical model $\mathcal{M}$ is used to provide a large number of outputs. The analysis might be tempted to construct a single ANN able to match all the outputs of interest. Instead, the use of different ANN to predict individual output is suggested. It is important to note that this computational strategy does not influence the capability of the ANN to model correlations among the outputs.

2. Numerical Example

2.1. DLR AIRMOD Structure

The experimental data come from a replica of the well-known GARTEUR SM-AG19 benchmark structure built at the German Aerospace Centre (DLR) in Göttingen, Germany and known as AIRcraft MODel (DLR-AIRMOD), and it is shown in Figure 1. The corresponding NASTRAN solid finite element model is also shown in Figure 1.

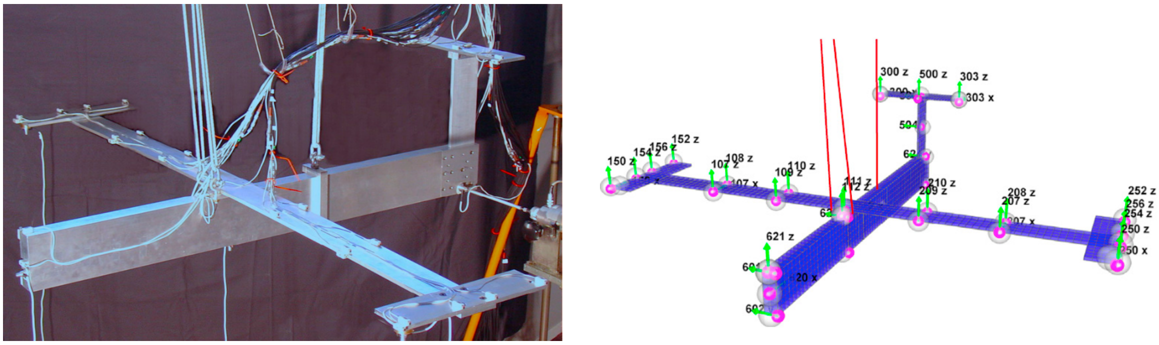

Fig. 1. DLR-AIRMOD structure [15].

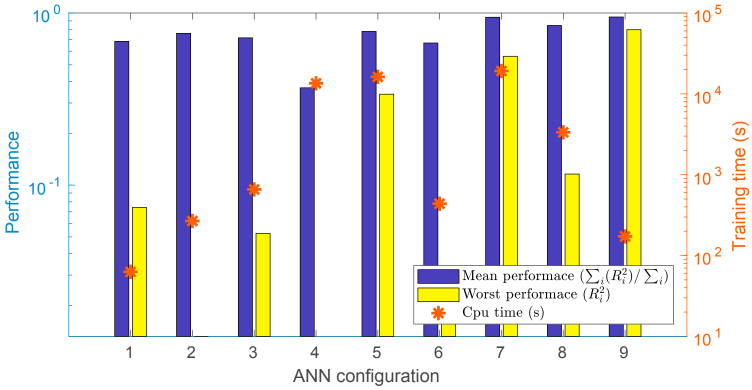

Fig. 2. Performance of different ANNs configurations and associated training time.

The experimental details are described in detail in [15]. The structure was reassembled 130 times, producing 260 modal data sets from single point excitation at two locations showing significant variability. 18 input parameters were selected including the support stiffnesses, joint stiffnesses, mass parameters to represent variability in the position of cable bundles, screws and glue after each reassembly of the structure and 14 natural frequencies are used as output of interest.

2.2. Analysis of Artificial Neural Networks

In this section, ANNs are constructed and trained to replace the computational expensive DLR-AIRMOD FE model and the performance of different ANN configurations is analysed. The training and validation data have been obtained using uniform distributions representing the variability of the 18 input parameters (from 5% till 200% of the nominal values). 2200 realisations of the input parameters have been obtained by means of the Latin Hypercube Sampling and split into two sets for calibration and validation.

9 different configurations of ANN have been tested as shown in Table 1. The first 8 ANNs are constructed to predict all the outputs of the DLR-AIRMOD FE model. Instead, the configuration #9 uses 14 individual ANN, each of those is trained to predict a specific output. The performance of the ANNs is then calculated evaluating the so-called R^2 parameter for each output of the ANN. The mean performance of the ANN is defined at the sum of each R^2 value divided by the number of considered outputs. The worst performance of the ANN correspond to the minimum value of the R^2. Figure 2 shows the performance of different ANN and the corresponding training time required. It can be seen that the best performance is obtained training individual ANN mapping all the inputs into a single output at time. The values for these configurations varying between 0.3677 and 0.9479 whereas the worst individual performance is between 0 and 0.7996. ANNs trained to represent single independent output (configuration #9) are outperforming all the ANNs trained to reproduce all the outputs at once while keeping the capability to reproduce output correlations (see Figure 2). Hence, the set of ANNs with single outputs is clearly the closest approximation of the DLR-AIRMOD

Table 1. An example of a table.

ID	Configuration (*t*)	Mean performance (*t*)	Worst performance (*t*)	Training time (s)
1	18 14 14	0.6840	0.0742	63
2	18 28 14	0.7613	0.0132	268
3	18 50 14	0.7175	0.0524	658
4	18 100 14	0.3677	0	13512
5	18 200 14	0.7818	0.3375	16218
6	18 28 24 14	0.6695	0.0227	4388
7	18 56 48 14	0.9446	0.5605	19173
8	18 40 36 14	0.8465	0.1160	3342
9	18 16 6 1	0.9479	0.7996	172

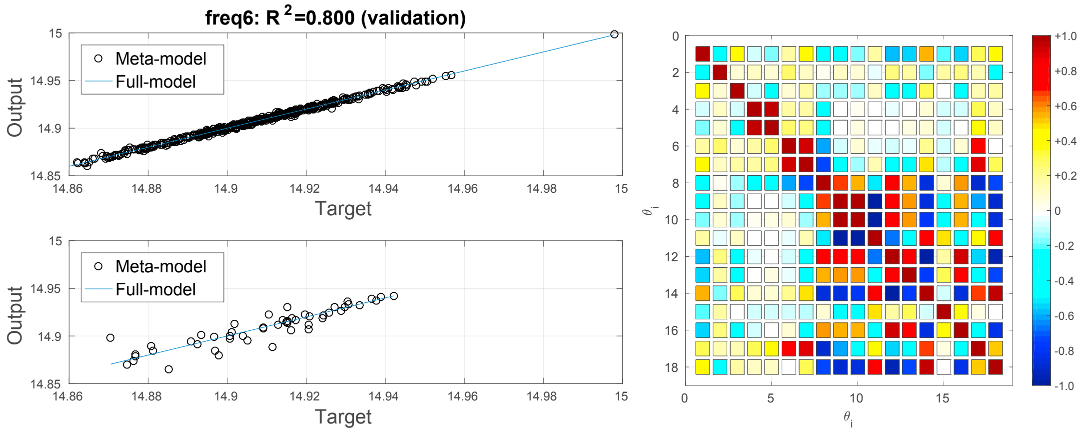

Fig. 3. Worst performance of the ANN configuration #9 (left panel) and predicted correlation among the outputs (right panel).

FE model. Also the training time (excluding the cost of running the full FE model 2000 times) for this configuration was found to be 172 seconds of CPU time, whereas the other configuration (which allowed for correlation between the 14 outputs) required up to 19,173 seconds (5.33 hours). The specific configuration of multi-input single output ANN has been estimated adopting heuristic search as shown in [9]. Auto-correlation of the vectors (of 2000 training points + 200 separately chosen validation points) for each of the 14 outputs were found to be in almost perfect agreement for the FE model and the ANN. The performance of the worst output prediction from the configuration #9 of ANN is shown in Figure 3.

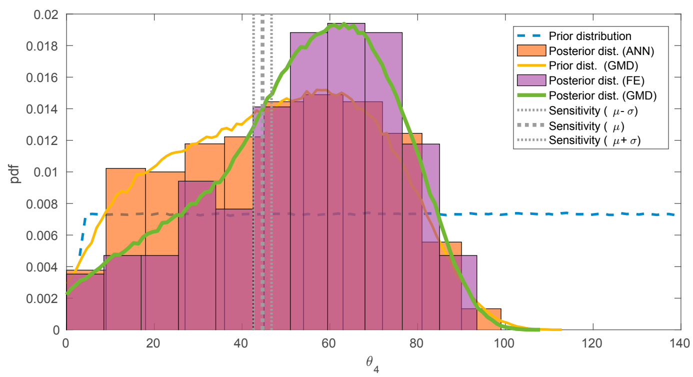

Fig. 4. Selected DLR-AIRMOD parameter updated using high-fidelity meta-model and FE model.

The selected ANNs has been used in the Bayesian Model and a sample example is shown in Figure 4. Those shown in orange were obtained directly from the ANN surrogate and the smooth orange curve is obtained from a Gaussian mixture based on 500 points determined by LHC sampling. The uniform prior is indicated by the horizontal dashed blue line and the vertical grey dotted lines denote the mean and two STDs determined by the sensitivity method (see [1] for more details about model uprating based on sensitivity method). Considerable confidence can be attributed to these results because the peaks of the orange histograms are seen in general to agree quite closely with the means obtained independently by the sensitivity method.

The presented procedures and numerical techniques have been implemented in the open source software OpenCOSSAN [16,17] and freely available to researchers and practitioners.

3. Conclusions

The use of a surrogate models enables efficient stochastic model updating, but at the same time introduces additional uncertainty. High-fidelity surrogate models are therefore required. In this paper 18 ANNs have been constructed to simulate the frequency distributions of the DLR-AIRMOD structure without loosing the correlation among the outputs. Such strategy has been proved to be more reliable and accurate than using a single ANN with multiple outputs. The availability of high fidelity surrogate models allows the implementation of stochastic model updating on large scale complex models allowing taking into account the unavailable variability of parameters.

References

[1] 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. doi:10.1007/s00419-017-1233-1.

[2] E. Patelli, D. A. Alvarez, M. Broggi, M. de Angelis, Uncertainty management in multidisciplinary design of critical safety systems, Journal of Aerospace Information Systems 12 (2015) 140–169.

[3] 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.

[4] Y. Yang, Y. Chen, A new direct method for updating structural models based on measured modal data, Engineering Structures 31 (2009) 32–42.

[5] 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.

[6] J. L. Beck, L. S. Katafygiotis, Updating models and their uncertainties. i: Bayesian statistical framework, Journal of Engineering Mechanics 124 (1998) 455–461.

[7] L. S. Katafygiotis, J. L. Beck, Updating models and their uncertainties. ii:model identifiability, Journal of Engineering Mechanics 124 (1998) 463–467.

[8] Z. Waszczyszyn, M. Slonski, Advances of Soft Computing in Engineering, Springer Wien - New York, 2010, pp. 237–316.

[9] U. Oparaji, R. Sheu, M. Bankhead, J. Austin, E. Patelli, Artificial neural network uncertainty quantification for the sensitivity analysis of the sixep model, in: 13th International Conference on Probabilistic Safety Assessment and Management (PSAM 13), 2016, p. 8.

[10] R. Schöbi, B. Sudret, S. Marelli, Rare event estimation using polynomial-chaos kriging, ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering (2016) D4016002.

[11] F. A. DiazDelaO, S. Adhikari, Gaussian process emulators for the stochastic finite element method, International Journal for Numerical Methods in Engineering 87 (2011) 521–540.

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

[13] Y. Govers, H. Haddad Khodaparast, M. Link, J. E. Mottershead, Stochastic model updating of the dlr AIrmod structure, Vulnerability, Uncertainty, and Risk (2014).

[14] J. Ching, Y. Chen, Transitional markov chain monte carlo method for bayesian model updating, model class selection, and model averaging, Journal of Engineering Mechanics 133 (2007) 816–832.

[15] Y. Govers, H. Haddad 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. doi:10.1061/9780784413609.049.

[16] E. Patelli, COSSAN: A Multidisciplinary Software Suite for Uncertainty Quantification and Risk Management. in: Springer International Publishing, Cham, 2016, pp. 1–69. doi:10.1007/978-3-319-11259-6_59-1.

[17] E. Patelli, M. Broggi, d. Marco, M. Beer, OpenCossan: An efficient open tool for dealing with epistemic and aleatory uncertainties, in: Vulnerability, Uncertainty, and Risk, American Society of Civil Engineers, 2014, pp. 2564–2573. URL: doi:10.1061/9780784413609.258.