Approximate Bayesian Computation for Probabilistic Damage Identification

Cecilia Viscardi[a], Silvia Monchetti[b], Luisa Collodi[a], Gianni Bartoli[b], Michele Betti[b], Michele Boreale[a], and Fabio Corradi[a]

[a]University of Florence - DiSIA, Viale Morgagni, 65, Florence; cecilia.viscardi@unifi.it, luisa.collodi@unifi.it, michele.boreale@unifi.it, fabio.corradi@unifi.it

[b]University of Florence - DICEA, via di S. Marta, 3, Florence; silvia.monchetti@unifi.it, gianni.bartoli@unifi.it, michele.betti@unifi.it

Abstract

Damage identification analyses are fundamental to guarantee the safety of civil structures. They are often formalised as inverse problems whose solution ignores any source of uncertainty that could be accounted for by using appropriate statistical models. Unfortunately, these models often exhibit an intractable likelihood function. We propose quantifying uncertainty through a fully Bayesian approach based on Approximate Bayesian Computation (ABC), a class of methods that overcome the evaluation of the likelihood and only require the ability to simulate from the model. Furthermore, we suggest a strategy to reduce ABC computational burden using Neural Networks. Finally, we test the method at work on a damaged beam to discuss its strengths and weaknesses.

Keywords: Damage Identification, Uncertainty Quantification, Approximate Bayesian Computation, Neural Networks

1. Introduction

In the civil engineering field, structural monitoring and damage detection techniques have received growing attention since they are paramount for tasks of control and preservation. Damages modify the mechanical properties of a civil structure and can cause changes in the dynamic behaviour of the system described by natural frequencies and modal shapes. Thus, these quantities can be exploited to infer the existence of structural damages, their location and their entity. A common practice is addressing the issue as an inverse problem: optimal values of the parameters describing the properties of the system are found by minimising a distance measure between the experimental data (e.g., observed frequencies) and data produced by a Finite Element Model (FEM) - i.e. a numerical method for solving the differential equations that describe the dynamic behaviour of the system as a function of the mechanical parameters and the structural configuration. Once solved the inverse problem, one can evaluate whether there have been variations of the mechanical properties carrying pieces of information on the location and the entity of the damage. However, such a procedure ignores all the sources of uncertainty - e.g. unobserved characteristics of the system, variations of the material properties as well as measurement errors. It follows that predictions about future dynamics are taken as assured. A probabilistic damage assessment allows for taking into account different sources of uncertainty. More specifically, a Bayesian probabilistic damage identification procedure provides posterior distributions over the location and the entity of the damage, thus avoiding a false sense of confidence. Moreover, posterior predictive distributions enable an evaluation of the uncertainty around the prediction of the future dynamic behaviour.

In the literature, there are few works addressing the problem of incorporating uncertainty in damage identification (7; 9, among others). They rely on strong assumptions and describe the relationship between observed data and mechanical properties through simple models that imply tractable likelihood functions. Implementing likelihood-free methods is a possible strategy to provide a finer description of reality. They allow a straightforward integration of the uncertainty induced by latent variables and variables having complex dependence structures. In (4; 3) the authors resort to Approximate Bayesian Computation (ABC), a likelihood-free approach, however, they do not adopt a fully Bayesian perspective and aim only at finding point estimates of the model parameters.

This paper is aimed at giving a formal statistical definition of the probabilistic model for damage's location identification building upon a proper framework for uncertainty quantification (5; 2). Furthermore, we describe a strategy to get fully Bayesian estimates using a suitable ABC algorithm. In particular, we propose a procedure based on a surrogate generative model derived via Neural Networks. We speculate that this approach will provide a flexible tool that allows straightforward integration into the model of many sources of uncertainty.

2. Bayesian Inference in models for uncertainty quantification

Let us denote by θ the variable object of our inference, that is the location of a damage in a structure, and by y_0 some observed characteristics related to its dynamic (e.g., the frequencies). Our aim is to derive the posterior distribution $\pi(\theta \mid y_0) \propto \pi(\theta)p(y_0 \mid \theta)$, given the prior distribution, $\pi(\cdot)$, and the likelihood function, $p(\cdot \mid \theta)$. [1] In this framework, the evaluation of posterior quantities requires a simulated inference approach because many unobserved variables interact with the damage's location and affects its relation with the frequencies. These variables must be included in the model as latent variables.

Let x be the latent variables and $\xi = (\theta, x)$ the vector of all the unknown quantities. In principle, Monte Carlo (MC) or Markov Chain Monte Carlo (MCMC) methods allow us to get samples from a posterior distribution defined on an augmented space: $\pi(\xi \mid y_0) = \pi(\theta, x \mid y_0) \propto \pi(\theta, x)p(y_0 \mid \theta, x)$. However, here even the likelihood function $p(y_0 \mid \theta, x)$ is analytically intractable and its evaluation may be computationally demanding. To give insights into the reasons for this intractability, we provide a formal statistical definition of the model for uncertainty quantification (2). The key elements of the model are:

- $y^R(\theta)$: the vector of real values of the frequencies when the damage's location is θ;
- $y^M(\xi)$: the output of a simulator that reproduces the real process. The simulator may be a numerical model for partial differential equations (e.g. the FEM) and takes both θ and x as inputs;
- $b(\xi) = y^R(\theta) - y^M(\theta, x)$: the discrepancy between the model and the reality. It may come from incorrect or missing physical characteristics, as well as the simplification of the problem needed to put it in a digital framework (e.g. space discretisation in FEM).
- $y^E(\xi) = y^M(\xi) + \eta(\xi)$: the emulator. It is an approximation of the simulator and $\eta(\xi)$ is the discrepancy between the simulator and the emulator.
- $y_0(\theta) = y^M(\xi) + b(\xi) + e$: the observed data. They typically differ from the real process for some measurement errors e.

In this scenario, the probability $p(y_0 \mid \xi)$ can be retrieved from $p(y_0, b, e \mid \xi)$ via marginalisation:

$$p(y_0 \mid \xi) = \int\int p(y_0 \mid e, b, \xi)p(e, b \mid \xi)de\, db = \int\int \delta_{y_0}(y^M + b + e)p(e)p(b \mid \xi)de\, db \qquad (1)$$

where $\delta_{y_0}(\cdot)$ is the Dirac measure. Note that Eq (1) comes from the assumption that measurement errors

[1]For the sake of simplicity, our notation does not discriminate between probability density functions and mass functions that can be distinguished from the context.

are independent of ξ and b, and from the fact that the simulator is a deterministic numerical model that, once a vector ξ is given as input, always returns the same output $y^M(\xi)$.

MC and MCMC algorithms for the computation of the $\pi(\xi \mid y_0)$ would involve multiple point-wise evaluations of $p(y_0 \mid \xi)$ and each of them requires the solution of the integrals in Eq (1). The computation of $y^M(\xi)$ makes exact calculations infeasible and numerical approximations are computationally demanding: a single evaluation of the integrals would require many runs of the FEM. This motivates the choice of simulation-based methods, such as Approximate Bayesian Computation (ABC).

3. ABC for probabilistic damage identification

The origin of ABC methods can be traced back to (11; 8) but, in the last twenty years, huge progress has been made in this field. For a comprehensive description of the method, we refer the reader to (10). The key idea of the basic ABC algorithm is to get samples from an approximate posterior distribution by converting samples from the prior into samples from the posterior in three steps: 1) generate N parameter values from the prior distribution $\pi(\cdot)$; 2) generate simulated processes $y_i \sim p(\cdot \mid \theta_i)$ for $i \in \{1, ..., N\}$; 3) retain only parameter values θ_i such that $d(y_i; y_0) \leq \epsilon$, where $d(\cdot\,;\,\cdot)$ is a distance function and $\epsilon \geq 0$ is a tolerance threshold.

The algorithm avoids the evaluation of the likelihood function. It only requires the ability to produce samples from $p(\cdot \mid \theta)$ using a *generative model* that can be thought of as a computer code which takes parameters θ as inputs, performs stochastic calculations that involve latent variables x, and outputs simulated data y. However, this solution to the problem of the intractability of the likelihood comes at the cost of introducing at least one source of approximation in the estimate of the posterior distribution. In particular, the quality of the approximation depends on the tolerance threshold ϵ: the approximate posterior distribution converges to the true posterior distribution as $\epsilon \to 0$.

Besides the basic ABC algorithm, many other sampling schemes have been proposed – see (10, Ch 4). Here, we resort to a sampling scheme inspired by the Population Monte Carlo ABC (PMC-ABC) presented in (1). It is displayed in Algorithm 1 where $K_j(\cdot \mid \theta_i^{j-1})$ is a Normal distribution with mean θ_i^{j-1} and variance equal to twice the weighted empirical variance of $(\theta_1^{j-1}, ..., \theta_N^{j-1})$, and α is a tuning parameter between 0 and 1. The output of the algorithm is a sample from the following approximate posterior distribution

$$\pi_\epsilon(\theta \mid y_0) = \pi(\theta) \int \mathbb{1}\{d(y; y_0) \leq \epsilon\} p(y \mid \theta) dy$$

where $\mathbb{1}\{\cdot\}$ denotes the indicator function and $\epsilon = \epsilon_M$ is the value of the threshold adaptively chosen.

Algorithm 2 describes the generative model used to get samples from $p(\cdot \mid \theta)$. Note that each run of the generative model involves a call to the FEM. Here we propose a strategy to reduce the computational cost of this procedure. In particular, we replace the simulator with a less expensive emulator based on Neural Networks.

Algorithm 1 ABC-PMC

Sample $\theta_1^0, ..., \theta_N^0$ from $\pi(\cdot)$.
Sample y_i using Alg. 2 giving θ_i^0 as input for each $i \in \{1, ..., N\}$.
Let $d_i = d(y_i; y_0)$ for each $i \in \{1, ..., N\}$.
Put ϵ_1 equal to the α-quantile of the distribution of $(d_1, ..., d_N)$.
for $j = 1, 2, \ldots, M$ **do**
  Set $i = 0$
  while $i < N$ **do**
    Sample θ^* from $q_j(\cdot) = \frac{\sum_{i=1}^N w_i^{j-1} K_j(\cdot|\theta_i^{j-1})}{\sum_{i=1}^N w_i^{j-1}}$.
    Sample y^* using Alg. 2 giving θ^* as input.
    Compute $w^* = \frac{\pi(\theta^*)}{q_j(\theta^*)} \mathbb{1}\{d(y^*; y_0) \le \epsilon_j\}$.
    if $w^* > 0$ **then**
      Let $\theta_i^j = \theta^*$, $d_i^j = d(y^*; y_0)$, $w_i^j = w^*$ and $i = i + 1$.
    end if
  end while
  Put ϵ_{j+1} equal to the α-quantile of the distribution of $(d_1^j, ..., d_N^j)$.
end for

Algorithm 2 Generative model

Take θ as an input.
Sample x from its prior distribution.
Compute $y^M(\theta, x)$ using the simulator (FEM) or the emulator (ANN).
Sample $b(\xi)$ and e from their distributions.
Return $y = y^M(\xi) + b(\xi) + e$.

Approximating the simulator using Neural Networks Artificial Neural Networks (ANN) are computational models that use experience to learn functions. We resort to *feedforward neural networks* which process information from inputs to outputs through intermediate computations and without feedback connections. The basic processing unit of an ANN is the *neuron* that receives inputs from other neurons and computes its own output using a linear combination based on previously defined weights and bias, and an *activation function*. Neurons are then arranged in layers: the first layer is called the input layer, the last layer is the output layer and in between layers are the hidden layers that determine the depth of the network. The network learns by adjusting weights and biases to minimise the prediction error, which is the value of the loss function that quantifies the difference between the output of the network and the output taken from a training set.

In the case of the probabilistic damage identification, the relation that links the unknowns, ξ, to the simulated process, y^M, is specified by a deterministic function reproduced using the FEM. Thus, an ANN can be used as an emulator that replaces FEM in Algorithm 2. The ANN takes as input layer ξ and gives as output layer an emulated process $y^E(\xi)$. To train the ANN we need a training set built by considering S vectors, $\xi^1, ..., \xi^S$, giving them as input to the FEM, and taking S simulated process as output. It follows that our approach gives a computational advantage as long as S is smaller than the number of simulations from the generative model in Algorithm 1.

4. Simulation study

To illustrate the proposed method, we considered the example of a simply supported beam modelled by using the commercial code ANSYS®. The total length of the structure is 5000 mm and the cross-section is a IPE240 steel profile – see Figure 1.

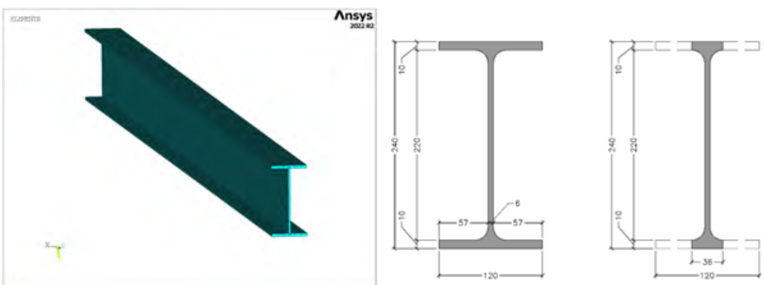

Figure 1: Discretisation in ANSYS®(left) and cross-section (right) of the beam without and with damage.

The induced localised damage is imposed by cutting the beam flanges. This damage is reproduced in ANSYS®by reducing the section shown in Figure 1. The effects, in terms of observable dynamic characteristics of the beam, are a reduction of the frequencies and changes in modal shapes. Here, we want to infer the location of the damage, $\theta \in (0, 5000)$, using three observed frequencies, $y_0 = (f_1, f_2, f_3)$ expressed in Hz. The latent variable X represents the uncertainty on the restraint conditions of the beam, in particular on its location. We assumed $\theta \sim \text{Uniform}(0, 5000)$ and $\frac{X}{100} \sim \text{Exponential}(\lambda = 1.5)$. Measurement errors $e = (e_1, e_2, e_3)$ are distributed as a Multivariate Normal with mean $\mu_e = (0, 0, 0)$ and covariance matrix $\Sigma_e = 0.15^2 I_3$. We included in the model also a random discrepancy $b \sim \text{Uniform}(-0.2, 0.2)$.[2]

In our simulation study, the observed data $y_0 = (32.63, 96.73, 208.61)$ have been produced running the FEM assuming $\theta^{\text{true}} = 2423.8$ and $x^{\text{true}} = 6.18$ (values generated at random). We trained the ANN with 2 inputs (θ and x), three hidden layers with 50 neurons, and 3 outputs (f_1, f_2 and f_3). We used the Relu activation function for all the layers, the Mean Square Error (MSE) as loss function, and Adam (6) as the optimization algorithm. Our training set has size $S = 160\,000$. The performance of the trained network is good enough to consider negligible the discrepancy between the simulator and the emulator (MSE $= (1.09 \cdot 10^{-5}, 4.88 \cdot 10^{-6}, 9.48 \cdot 10^{-6})$ and $R^2 = (0.99, 0.99, 0.99)$ computed on a test set).

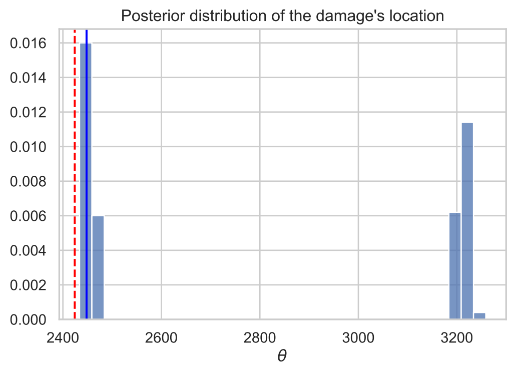

Figure 2: Posterior distribution of θ with θ^{true} (red line) and θ^{MAP} (blue line).

	Posterior predictive distributions		
	mean	s.d.	95% CI
f_1	38.60	2.22	32.88; 41.03
f_2	113.31	6.26	97.70; 120.85
f_3	243.56	14.41	210.60; 263.30

Table 1: Mean, standard deviation and 95% credible intervals of the posterior predictive distributions.

We ran Algorithm 1 for 20 minutes with $N = 500$ and using the Euclidean distance. In the given budget of time, the final number of iterations is $M = 22$, and all of them required more than 100 000 calls to the generative model to accept 500 parameter proposals. The final threshold is $\epsilon_M = 0.07$.

Looking at Figure 2 we can see that the Maximum a Posteriori estimate of the damage's location, $\theta^{\text{MAP}} = 2448$, is very close to θ^{true}. Note that the FEM discretises the beam using meshes of size 50 mm, meaning that the difference $\theta^{\text{MAP}} - \theta^{\text{true}} = 24.2$ mm can be ignored since it is too small to be detected by the model. The bimodality of the posterior distribution is due to the symmetry of the beam. However, in the reality this perfect symmetry does not occur, thus we speculate that posterior distributions based on real data will be unimodal. The uncertainty about the damage's location has been propagated to future frequencies by computing posterior predictive distributions described in Table 4.
The variability of the distribution increases moving from the first to the third frequency. In a more realistic framework, the uncertainty would be even larger and ignoring posterior predictive distributions may lead to the observation of completely unexpected scenarios.

5. Discussion and future work

In this work, we investigate the use of a formal model for uncertainty quantification in the identification of damages in civil structures. A probabilistic approach is essential to be aware of the uncertainty

[2]Prior distributions have been set exploiting information coming from preliminary investigations as well as the experts' knowledge.

around estimates and predictions and to conduct a more conscious process of decision-making. However, including different sources of uncertainty often leads to complex probabilistic models with an intractable likelihood function. We propose a likelihood-free approach to provide fully Bayesian estimates. The presented ABC method overcomes problems related to the computational cost of the simulator resorting to an emulator. In particular, we exploit the deterministic nature of the function that links parameters and latent variables to the frequencies and propose an emulator based on ANNs.

Our exploratory analysis showed that the method is able to infer the damage's location and gave some insights into the uncertainty of future frequencies pointing out the importance of considering posterior predictive distributions. This aspect makes the proposed framework particularly relevant in the structural health monitoring field.

One of the main strengths of the proposed approach is its flexibility. In our example, we assumed simple Gaussian and Uniform distributions over the measurement errors and the bias of the model. However, the method allows one to straightforwardly replace them with more complex random variables – e.g. considering Gaussian processes (5). In fact, we need only the ability to produce simulations from the assumed distributions and no analytical evaluations are required. Furthermore, in this framework, the statistician can take full advantage of the expert knowledge in the specification of the prior distributions and the definition of a model that is as close as possible to reality.

The major drawback of the method is that it still requires a large number of calls to the FEM to build a training set for the ANN. However, in our example, the size of the training set turned out to be far lower than the number of simulations needed in the PMC-ABC procedure. Furthermore, the use of the trained emulator is not limited to the implementation of the PMC-ABC algorithm since it can be integrated into the health monitoring system which, otherwise, would call the FEM many times.

Future work should focus on the application of the method to real data and on the extension to more complex structures such as bridges, towers, etc. Moreover, we plan to define a more sophisticated model that allows inferring also the presence/absence and the entity of the damage.

References

[1] M. A. Beaumont, J.-M. Cornuet, J.-M. Marin, and C. P. Robert. Adaptive approximate bayesian computation. *Biometrika*, 96(4):983–990, 2009.

[2] J. O. Berger and L. A. Smith. On the statistical formalism of uncertainty quantification. *Annual review of statistics and its application*, 6:433–460, 2019.

[3] S.-E. Fang, S. Chen, Y.-Q. Lin, and Z.-L. Dong. Probabilistic damage identification incorporating approximate bayesian computation with stochastic response surface. *Mechanical Systems and Signal Processing*, 128:229–243, 2019.

[4] Z. Feng, Y. Lin, W. Wang, X. Hua, and Z. Chen. Probabilistic updating of structural models for damage assessment using approximate bayesian computation. *Sensors*, 20(11):3197, 2020.

[5] M. C. Kennedy and A. O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.

[6] D. P. Kingma and J. Ba. Adam: a method for stochastic optimization 3rd int. In *Conf. for Learning Representations, San*, 2014.

[7] Y. Lei, Y. Su, and W. Shen. A probabilistic damage identification approach for structures under unknown excitation and with measurement uncertainties. *Journal of Applied Mathematics*, 2013.

[8] J. K. Pritchard, M. T. Seielstad, A. Perez-Lezaun, and M. W. Feldman. Population growth of human y chromosomes: a study of y chromosome microsatellites. *Molecular biology and evolution*, 16(12):1791–1798, 1999.

[9] R. Rocchetta, M. Broggi, Q. Huchet, and E. Patelli. On-line bayesian model updating for structural health monitoring. *Mechanical Systems and Signal Processing*, 103:174–195, 2018.

[10] S. A. Sisson, Y. Fan, and M. Beaumont. *Handbook of approximate Bayesian computation*. CRC Press, 2018.

[11] S. Tavaré, D. J. Balding, R. C. Griffiths, and P. Donnelly. Inferring coalescence times from dna sequence data. *Genetics*, 145(2):505–518, 1997.