A NOVEL BLIND PARAMETRIZATION ALGORITHM FOR DAMAGE DETECTION IN CFRP PLATES

by

Sergio Cantero Chinchilla

A thesis submitted to the Department of Structural Mechanics and Hydraulic Engineering,

in partial fulfillment of the requirements for the degree of

MSc. IN STRUCTURAL ENGINEERING

Supervisor: Dr. Guillermo Rus Carlborg

Department of Structural Mechanics and Hydraulic Engineering

University of Granada, Campus de Fuentenueva,

18071 Granada, Spain

July 2014

ABSTRACT

A NOVEL BLIND PARAMETRIZATION ALGORITHM FOR DAMAGE DETECTION IN CFRP PLATES

A novel blind damage parametrization algorithm is presented for selecting the most plausible parametrization to identify damage in ultrasonic measurements obtained from a damaged carbon fiber-reinforced polymer plate. This parametrization is selected among all possible ones, extracting those whose plausibilities are higher, so the ultrasonic signals can be well fitted. The algorithm performance is validated with a set of synthetically damaged signals. The results show that the algorithm is able to detect the right damage positions and amounts, even if the added level of white Gaussian noise less than a signal-to-noise ratio of 25 dB. In addition, some experimental signals obtained from a post-impact fatigued CFRP plate were evaluated with our algorithm under two different damage configurations. The first one, which entails stiffness degradation of the layers, results as more plausible than the second one, which entails stiffness degradation and attenuation increase. The obtained damage patterns were compared to classic observational techniques such as micrography.

Keywords: Bayesian inverse problem, Ultrasonic NDE, US modeling, Carbon fiber-reinforced polymer, post-impact fatigue damage, blind parametrization algorithm, damage detection and reconstruction.

RESUMEN

UN NOVEDOSO ALGORITMO DE PARAMETRIZACIÓN PARA LA DETECCIÓN DE DAÑO EN LAMINADOS DE CFRP

Un novedoso algoritmo de parametrización de daño se presenta para seleccionar la parametrización más plausible que pueda identificar daño en medidas ultrasónicas obtenidas de un laminado de fibra de carbono. Esta parametrización se elige de entre todas las posibles, extrayendo aquellas cuyas plausibilidades son mayores, de tal manera que las señales ultrasónicas puedan ser mejor ajustadas. La validación del algoritmo se presenta con un set de señales sintéticamente dañadas. Los resultados muestran que el algoritmo es capaz de detectar correctamente la posición y cantidad de daño, incluso si el nivel de ruido blanco Gaussiano añadido a la señal llega hasta un ratio de señal-ruido de 25 dB. Además, algunas señales experimentales escogidas del especimen dañado con fatigua post-impacto fueron introducidas en nuestro algoritmo con dos configuraciones diferentes. La primera, que implica una degradación de la rigidez de las capas, se mostró como más plausible que la segunda, la cual implica una degradación de la rigidez además de un incremento de la atenuación. Los patrones de daño obtenidos fueron comparados con las técnicas observacionales clásicas como la micrografía.

Palabras clave: Problema inverso bayesiano, END ultrasónica, modelos US, fibra de carbono, daño de fatiga post-impacto, algoritmo de parametrización ciega, detección y reconstrucción de daño.

ACKNOWLEDGMENTS

I would like to thank the responsible for the direction of my research, Dr. Guillermo Rus Carlborg of the Department of Structural Mechanics. His philosophical thinking has a great influence on my work throughout this time. I could no forget to thank my colleges of the Nondestructive Evaluation Laboratory, in particular to Juan Chiachio Ruano, Nicolas Bochud and Manuel Chiachio Ruano. There have been a lot of enjoyable moments in the collaborations and I have learned much from them, especially in the areas outside my research topic. Also I would like to thank my colleges of the Department of Structural Mechanics for their friendly help and advise.

And finally, I need to express my sincere gratitude to my family in addition to the person who was beside me. I'm in debt with them for the comprehension I receive in my MSc work.

Contents

Chapter 1

Introduction

1.1 Context and motivation

Fiber-reinforced polymer (FRP) materials have been widely used in aeronautics applications, and are continuously growing in civil engineering, since they show a higher performance (i.e. strength and durability) than classic materials like steel [1]. For example, these materials have a higher resistance when loaded in fiber direction [2], when subjected to cyclic mechanical loading (or fatigue loading). Nevertheless, they suffer from limitations since they may be affected by some different types of damage. Some of the most known damage are construction defects (for instance, micro-bubbles within the resin of the material, called matrix or debonding fiber-resin), extreme environmental changes (i.e. contractions and dilatations due to thermal amplitude), impact damage, which can be the responsible for more severe damage since this material does not allow dissipation by plastic deformation in the thickness direction [3], or fatigue damage, which can produce matrix cracks, stiffness degradation or fiber-matrix debonding [4]. Alternatively, post-impact fatigue damage, which is one of the most complex damages in composites, such as carbon fiber-reinforced polymer (CFRP), has been widely studied in the literature trying to avoid harmful effects in the FRP plates [5, 6]. However, damage growth mechanisms on fatigue after impact damage are not fully understood yet [7]. Consequently, nondestructive evaluation (NDE) techniques have been developed to detect such damage within the specimens. One of the most frequently used NDE techniques for this purpose is ultrasound (US) [8, 9].

Ultrasonic signals obtained from materials put down post-impact fatigue loading have a high complexity. Therefore, a reliable ultrasonic wave propagation (UWP) model that can extract relevant damage information from those signals is desirable. To this end, a model-based inverse problem (IP) can be applied to reconstruct the values of the damage parameters that fit best the experimental measurements. The theoretical background on the inverse problem theory is provided by Tarantola [10]. The fundamental idea of the IP consists of using an iterative strategy based on the minimization of the discrepancy between the experimental and the modeled signal responses. Next section reviews some proposals of the literature that have been developed within this context

1.2 Literature review

The reconstruction of the mechanical properties of damaged composites from US measurements, such as layered CFRP's, has scarcely been addressed by solving IP's. Thus, some of these reviews account for multilayered materials, as a generalization for composite materials. In that vein, Kinra and Zhu [11] developed an inverse algorithm, which utilizes the well-known Newton-Raphson method, to reconstruct the thickness and the phase velocity of a thin coating on a thick substrate, through a comparison of the theoretical and the measured transfer functions. In a similar proposal, Kinra et al [12] developed another inverse algorithm, which utilizes either the Newton-Raphson or the Simplex method in conjunction with the incremental search method in order to reconstruct simultaneously the thickness and the phase velocity of each individual layer. These results are obtained by minimizing the difference between the experimental and the theoretical results in a mean-sum-square sense. Instead of comparing the time-domain signals, they consider the phase spectrum in the IP. In another related proposal, Kinra and Iyer [13] presented an inverse algorithm, which utilizes the secant method [14], and could estimate one of the following parameters given the remaining three: thickness, wave speed, density and attenuation. This was possible through a comparison between the experimentally obtained and theoretically predicted transfer function.

Alternatively, Balasubramaniam and Whitney [15] presented an inverse technique for computing the material elastic constant from data acquired with an immersion through-transmission method for characterizing thick glass-epoxy composites. In addition, Balasubramaniam and Rao [16] described an inverse technique based on genetic algorithms that reconstructed the material stiffness properties. In a similar approach, Fahim et al [17] proposed a model-based inverse problem for detecting variations in structural parameters (stiffness properties and attenuation coefficients) due to impact damage in composites. Genetic algorithm was used for the parameter search.

As a drawback, most of these works are applied to determine the properties of a limited amount of layers and have limitations such as uncertainties derived from the measurement techniques (i.e. errors in the transmitter-receiver alignment, bad positioning of the specimen, etc.), noise in the ultrasonic signals which can produce distortions in the shape of the wave and modify the parameters reconstruction, or just from the numerical method like gradient-based algorithms which do not attain global convergence [17]. In addition, they introduced a parametrization (the group of parameters that can vary) as invariant, so they intrinsically assume those parametrizations as the best. To face these problems, we propose a Bayesian inverse problem (BIP), which enables us to incorporate the uncertainties from the measurments noise and that of the model within the inversion scheme. Bayes' Theorem is applied at two levels: (1) to deal with the posterior information about the UWP model parameters for a specific parametrization (model-class), and (2) to assess a degree of plausibility of each model-class within a set of them [18]. This framework has been recently used to assess fatigue models in CFRP material by Chiachio et al, since it takes into account the observation and model uncertainties in application to the problem of fatigue degradation [19, 20]. Despite all these contribution, there are still voids in this field that we try to fill in with our proposal.

1.3 Research objectives

Damage identification in multilayered materials is a challenging problem. To this end, the ultrasonic nondestructive evaluation is chosen to detect and parametrize pathologies in these materials. In this work, a strategy based on Bayesian inverse problem (BIP) along with an ultrasonic wave propagation (UWP) model is adopted to develop a global tool that provides the most plausible parametrization, and then, the underlying damaged pattern that explains the data obtained from a damaged CFRP specimen. This is the final objective of the research presented in this thesis. To approach this objective, several concrete objectives are presented on the basis of some assumptions to validate or falsify, listed below.

- ◇ Post-impact fatigue damage in CFRP plates is not fully understood yet because of its complexity. Consequently, a parametrization considered as invariant can introduce erroneous results in the signal reconstruction, since the damage distribution may be *a priori* unknown. In addition, uncertainties derived from the noise in the ultrasonic signals and the measurement techniques make that the deterministic inverse problem can contaminate the signal and parameter reconstructions. →*Research assumption 1:* A blind parametrization algorithm can deal with the damage detection and the material properties reconstruction, by merging the BIP with the UWP model.

◇ The material properties reconstruction from ultrasonic signals measured in a damaged CFRP specimen can be addressed not only with a unique variable, but with several configurations, which are referred to different variables (i.e. stiffness modulus or attenuation coefficients), since a single type of damage can be explained with different variables. →*Research assumption 2:* A blind parametrization algorithm is expected to be able to reconstruct signals obtained from real inspection conditions, with two different configurations at the same level of reliability.

1.4 Outline of the thesis

This thesis intends to provide suitable solutions to the several research questions outlined in the previous Section. The methods and experiments designed to investigate these research questions are listed below:

Research question 1:

A blind parametrization algorithm can deal with the damage detection and the material properties reconstruction, by merging the BIP with the UWP model.

The description of the ultrasonic wave propagation (UWP) model used in this work is summarized in **Chapter 2**, Section 2.2. On the other hand, the Bayesian framework used in this algorithm is presented in Chapter 2, Section 2.4. Afterwards, the blind parametrization algorithm is depicted in Chapter 2, Section 2.5, and the MATLAB codes of the main algorithm and its subroutines are presented in the **Appendices A** and **B**. In addition, the reliability and robustness of the algorithm are proved by making use of several synthetic signals. The description of the synthetic signals is explained in Chapter 2, Section 2.3, whereas the results of the test with the signals and parameters reconstruction are presented in **Chapter 3**, Section 3.2. Moreover, the interpretation of these results is outlined in **Chapter 4**, Section 4.1.

Research question 2:

A blind parametrization algorithm is expected to be able to reconstruct signals obtained from real inspection conditions, with two different configurations at the same level of reliability.

The experimental evaluation of the proposed algorithm is carried out by making use of some ultrasonic signals measured in a damaged CFRP specimen. The material description and experimental procedure to take the measurements are depicted in Chapter 2, Section 2.1. In addition, the set of variables chosen to reconstruct the damage is explained in Chapter 2, Section 2.3. On the other hand, the algorithm specifications are presented in Chapter 3, Section 3.1 and the results of the experimental validation are presented in Section 3.3. The interpretation of these results is outlined in Chapter 4, Section 4.1. Finally, some concluding remarks and future trends based on this work complete the Chapter 4, Sections 4.1 and 4.2.

Chapter 2

Methodology

The proposed methodology consists of five steps. In first place, Section 2.1 briefly describes the experimental process used to manufacture the specimen and the ultrasonic setup employed to measure it. Secondly, Section 2.2 presents the wave propagation model used to idealize the experimentally obtained ultrasonic signals. Then, the Bayesian hypothesis testing is explained in Section 2.4, with emphasis on the Bayesian inverse problem, the Bayesian model-class and the Metropolis-Hastings algorithm used for Bayesian model updating. Finally, the blind algorithm developed to identify the damage is carefully outlined in Section 2.5.

2.1 Experimental setup

Ultrasonic signals used here were obtained from a specimen previously tested at the Nondestructive Evaluation Laboratory. This specimen was damaged with two types of tests (impact and fatigue). In order to detect the applied damage in the CFRP plate, ultrasonic signals were obtained in an immersion tank.

2.1.1 Material description

The CFRP specimen used in this work has been manufactured at the Institute of Polymers and Composites, TU Hamburg-Harburg, Germany. This specimen is compound by several layers of carbon fibers stacked with epoxy resin between them. The orientation of the fibers of each layer was selected in two perpendicular directions, which are called 90° and 0° directions in reference to the main directions of the laminate (denoted by $[0/90]_{4s}$, where 0/90 indicates the two main directions and the subscript $4s$ indicates the number of layers stacked together).

After the laminate construction, curing was applied. At this stage of manufacturing, the specimen was placed in autoclave at 177°C for three hours with a pressure of 7 bar. This process is necessary for the epoxi resin in order to react chemically and achieve its top mechanical properties like stiffness, strength, etc. The final dimensions of the specimen are $250 \times 35 \times 2$ mm^3.

Let us consider the well-known mixing rules [21], see Equation (2.1), which enable us to obtain the properties of each layer. To this end, it is necessary to count with the properties of the matrix (epoxy resin) and the volume fraction of fibers, which were provided by manufacturer.

$$\begin{aligned} E_{eff} &= \frac{1}{\dfrac{V_f}{E_f} + \dfrac{V_m}{E_m}} \\ \nu_{eff} &= V_f \cdot \nu_f + V_m \cdot \nu_m \\ \rho_{eff} &= V_f \cdot \rho_f + V_m \cdot \rho_m \end{aligned} \tag{2.1}$$

In Equation (2.1) E_{eff} is the effective Young's modulus of the layer, ν_{eff} the effective Poisson ratio of the layer, ρ_{eff} the effective density of the layer. The subscript f is referred to the fibre fraction whereas the subscript m to the matrix fraction. The volume fraction of each material in each layer is denoted by V. Thus, the effective properties of the layers that compose the CFRP plate are shown in Table 2.2.

Layer properties	Young modulus E $[GPa]$	Poisson ratio ν $[-]$	Density ρ $[kg/m^3]$	Attenuation α $[Np/m]$	Thickness a $[mm]$
0°/90°-orientation	11.1616	0.3007	1589.5	293.023	0.1215

Table 2.1 *Mechanical and geometrical properties of the layers that compose the CFRP plate.*

The specimen was damaged with two types of tests. At first, the CFRP plate was impacted with 3.8 Joule energy and after this, the specimen was put down fatigue test (tension and compression) up to 100000 cycles.

2.1.2 Experimental procedure

The ultrasonic technique used for capturing the measurements of this work is the normal incidence technique [22, 23], which allows us to detect damage parallel to the surface such as delaminations or matrix cracking when cracks do not appear perpendicular to the surface. These ultrasonic wave signals were obtained by Bochud [24] using an immersion through-transmission subwavelength technique as in [11]. The experimental procedure used to obtain those signals is briefly summarized hereafter.

The CFRP plate was excited by a low-frequency ultrasonic sine-burst at 5 MHz, which was produced by a wave generator (Agilent 33220). The specimen was placed at the focal distance ($d_f = 30$mm) between the focused transducers. Once recieved, the signals were amplified and digitized with an adquisition card, and then analyzed off-line in a personal computer. The response signals were measured on a damaged area corresponding to an extension of 40×20 mm^2 (with a 1 mm step).

This configuration provides us 861 measurements. In this work, only a

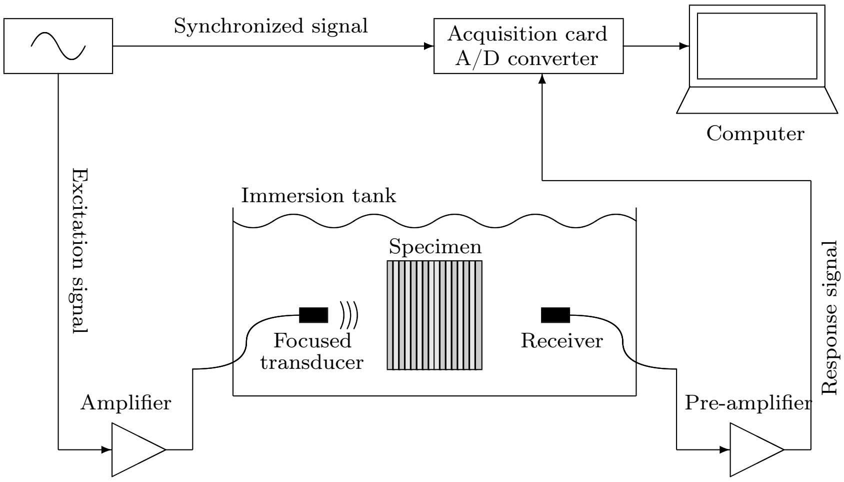


Figure 2.1 Experimental configuration of the excitation-propagation-measurement system[1]

few of these signals are used. They were carefully selected in order to give a representative view of different damage mechanisms (i.e. matrix cracks, delaminations, etc.)

2.2 Ultrasonic wave modeling

The ultrasonic wave propagation model used in this work in order to mimic the ultrasonic signals experimentally obtained is that of Bochud's thesis [24], because of its low complexity and efficiency; those characteristics are ideal to run a model-based probabilistic inverse problem (PIP) approach, since this kind of approach needs a huge number of evaluation of the forward problem.

This model is digital and relies on the properties of the CFRP layers: thickness, Young's modulus, density, damping and Poisson ratio (some of these properties are obtained with the mixing rules presented in Equation (2.1)). Once all these properties are introduced into the model, acoustic properties like wave velocity and impedance ratios (water and specimen) are generated and converted into model parameters, and later transformed

[1] Reproduced from Bochud [24].

into filter coefficients. This procedure is sketched in Figure 2.2. Thus, we have the filter parameters needed to generate a ultrasonic wave signal that idealizes the specimen and water properties. The aforementioned signal will be the center of the study to detect possible damages in each layer or interface.

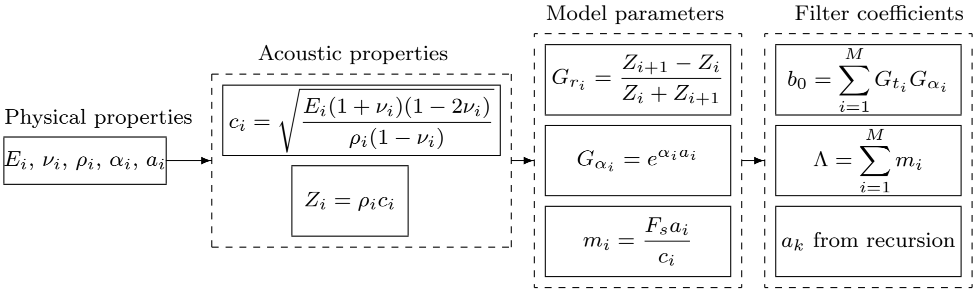

Physical properties
$E_i,\ \nu_i,\ \rho_i,\ \alpha_i,\ a_i$
Acoustic properties
$c_i = \sqrt{\dfrac{E_i(1+\nu_i)(1-2\nu_i)}{\rho_i(1-\nu_i)}}$
$Z_i = \rho_i c_i$
Model parameters
$G_{r_i} = \dfrac{Z_{i+1}-Z_i}{Z_i+Z_{i+1}}$
$G_{\alpha_i} = e^{\alpha_i a_i}$
$m_i = \dfrac{F_s a_i}{c_i}$
Filter coefficients
$b_0 = \sum_{i=1}^{M} G_{t_i} G_{\alpha_i}$
$\Lambda = \sum_{i=1}^{M} m_i$
a_k from recursion

Figure 2.2 Computational process to take layered material properties and convert them to a functional digital filter.[2]

The material idealization is not completed yet. The fact that there is no possibility to observe a delamination (one of the most common damage produced by impact in laminates) has to be taken into account, since it occurs between plies. To model this kind of damage it is necessary to assume a small interface layer, whose thickness is thinner than both plies and wavelength of the emitted ultrasonic wave. Since the manufacturing procedure of CFRP sample was made by stacking up each sequence of four layers, it is reasonable to consider that between each ply, there is a small interface layer of matrix, which in this case consists of epoxy resin. For this reason, epoxy resin properties (Table 2.2) were assigned to simulate those interfaces. Thus, the model, has $N = 33$ layers (that are the 16 layers of the real CFRP plate, plus 15 interfaces of epoxy resin in-between, plus 2 water layers ahead of and behind the specimen).

Given those properties (see Table 2.1 and 2.2), which include layers and interfaces properties, the UWP model can be computed straightforwardly,

[2] Reproduced from Bochud [24].

Interface properties	Young modulus E $[GPa]$	Poisson ratio ν $[-]$	Density ρ $[kg/m^3]$	Attenuation α $[Np/m]$	Thickness a $[mm]$
Interface	5.2728	0.3500	1310	361.1595	0.01

Table 2.2 *Properties of the interfaces assumed between every two consecutive layers in the digital model.*

and thus provides an approximation of a measurement from an undamaged area.

2.3 Damage hypothesis

One of the challenges that we face up in the damage reconstruction problem is to deal with such a large set of damage model parameters. In fact, any of these parameters $\boldsymbol{\theta} \in \{E_i, \nu_i, \rho_i, \alpha_i, a_i\}$ can be affected by damage, with $i = 1, \ldots, M$. All these parameters provide information about the mechanical and geometrical performance of the entire CFRP plate. In order to solve this problem with a reasonable computational cost, a reduced number of variables have to be selected. It can reasonably be assumed that damage affects mostly the global stiffness, reducing its nominal value, or the attenuation coefficient, increasing its nominal value. Thus, the Young modulus E or the attenuation coefficient α of each layer/interface were chosen as potential variables.

Post-impact fatigue damage is not fully understood yet [7], and the complex distribution of damage is not known a priori. To face this issue, we propose a blind damage parametrization that searches damage iteratively within any single layer i, with $i = 1, \ldots, M$, in order to obtain the best matching between the modeled and experimental signals. Once the first damage parameter is identified, it is stored and then the search begins again with the aforementioned parameter fixed plus the current one. For more information about the algorithm, see Section 2.5.

To evaluate the robustness of this novel blind damage parametrization algorithm, it is necessary to evaluate it with rather simple signals, whose damage distribution is known a priori. To this end, we generated several synthetically damaged signals by running the forward problem with our model and adding different level of white Gaussian noise (WGN) to it. An enclosed set of synthetic damaged signals has been generated with the following damage distribution (i.e. the numbers between brackets indicate the position of the damage layer/interface according to Figure 3.1):

Damage localization	Layers (Plies)	Interfaces (Matrix)
Test no. 1	[1]	[2 30]
Test no. 2	[19 29]	[2 18 30]
Test no. 3	[1 15 17]	[30]

Table 2.3 *Damage distribution to generate signals in order to validate the algorithm.*

Those sintethycally damaged signals were created so that the damage follows a realistic distribution, according to the description available in the literature [8, 9, 25]. For impact damage, effects are supposed to be in group of layers and interfaces opposite to the impacted side, in addition to the first layer and interface, which is responsible to endure high part of impact energy. Typical delaminations and matrix cracking in CFRP's can be seen in Figure 2.3. Besides, fatigue test spreads cracks and micro-cracks into 90° plies,and even delaminations can grow in extension to them.

In test number 1, damage in the first layer (where drop weight impacts firstly) and in two interfaces has been proposed: the first one, which can be produced by violent impacts and the last one, which is probably one of the most likely position for delaminations (see Figure 2.3). For test number 2, a more complex damage distribution has been chosen. Delaminations have been assumed as consequence of matrix cracking propagation. This could be explained because when cracks become larger, some delaminations

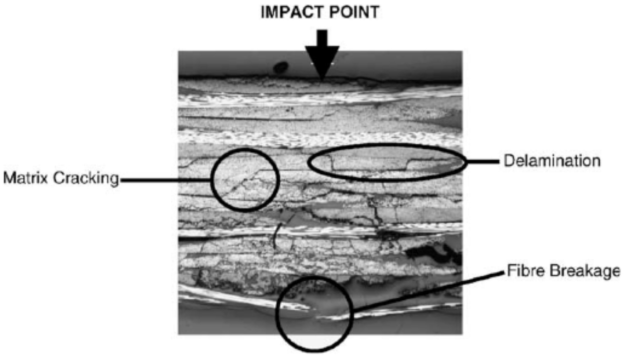


Figure 2.3 Micrograph of impact damage in composite sample[3].

can appear in the adjacent interfaces of layers of different orientation. In addition, we chose delamination in the first layer of the CFRP plate because of the same reason given for the test 1. Finally, a third test has been created with different kind of damage. One delamination stands in the last interface, as consequence of the impact, and the first layer is damaged for the same reason as before, but two central layers are additionally chosen, whose orientation is 90°, are damaged because of fatigue damage by tension and compression loading. The level of noise added to the signals has been addressed by computing the signal-to-noise ratio (SNR) [27] as,

$$SNR = 10 \log \left(\frac{E_s}{\sigma_\nu^2} \right) \quad [dB]. \tag{2.2}$$

expressed for white Gaussian noise (WGN). The SNR relates the level of input energy of the signal (E_s) with the variance of the WGN (σ_ν^2). Generally, three levels of noise were considered.

2.4 Bayesian hypothesis testing

Once the UWP model and the damage hypothesis have been defined, the assessment of the BIP has to be addressed. The hypothesis testing consists of evaluating one parameter in each layer and interface of the laminate

[3] Reproduced from Mitrevski et al. [26].

(Young's modulus or attenuation coefficients for each layer and stiffness modulus for each interface). Each single value of a parametrization produces an ultrasonic wave signal, so we have to test a huge number of values (by the forward problem) to achieve those which fit better the experimental signal. To face this problem, we use Bayesian methods (Figure 2.4) for model assessment, since they only use the axioms of probability [18, 28].

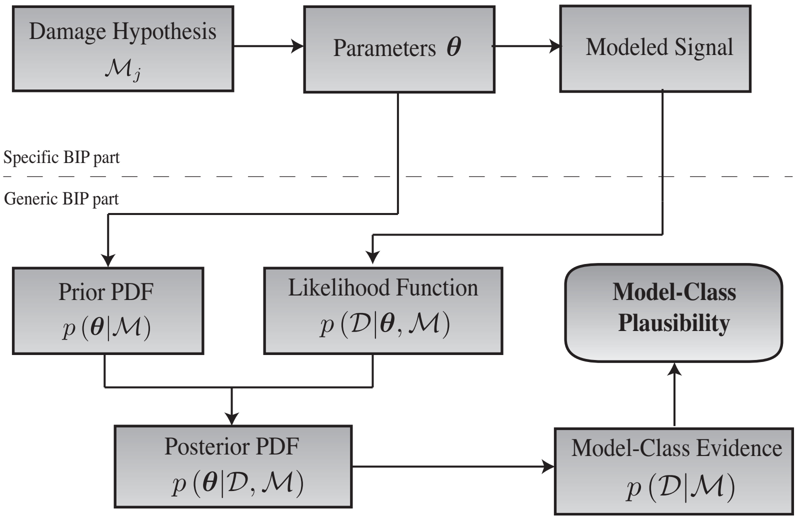

Damage Hypothesis
$\mathcal{M}_j$
Parameters $\boldsymbol{\theta}$
Modeled Signal
Specific BIP part
Generic BIP part
Prior PDF
$p(\boldsymbol{\theta}|\mathcal{M})$
Likelihood Function
$p(\mathcal{D}|\boldsymbol{\theta},\mathcal{M})$
Model-Class Plausibility
Posterior PDF
$p(\boldsymbol{\theta}|\mathcal{D},\mathcal{M})$
Model-Class Evidence
$p(\mathcal{D}|\mathcal{M})$

Figure 2.4 Bayesian framework applied for damage detection.

In order to make this Bayesian procedure more understandable, the process for assessing damage hypothesis described in Section 2.3 is depicted in Figure 2.4. Every step is described in the following subsections, with a general explanation of Bayesian inverse problem, the stochastic transformation of the modeled signal, the formulation of the likelihood function as well as the process to obtain the posterior probability density of function (PDF) and evidence of each model-class. The model-class $\mathcal{M}$ in this work is referred to the UWP model along with the underlying damage hypothesis done before, that are enclosed within a certain model parametrization.

2.4.1 Bayesian inverse problem

The way of assessing this problem is addressed with Bayesian inverse problem, since this type of inverse problem can better deal with noise, heterogeneity of properties within the CFRP plate and approximation of reality made by the UWP model, than a deterministic IP. Bayesian inverse problem provides information about how probable is that our model parameters $\boldsymbol{\theta}$ over the set $\boldsymbol{\Theta} \subset \mathbb{R}^d$ are fitted to data $\mathcal{D}$, which contain information about experimental tests and some important prior information on those parameters. This information is presented in form of the *posterior* PDF ($p(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M}_j)$), since this PDF shows how the plausibility of the values $\boldsymbol{\theta}$ are conditionated by experimental and prior data and by the model class $\mathcal{M}$. Experimental data $\mathcal{D}$ consist of a sequence of ultrasonic signals or synthetic signals that can be written as $\mathcal{D} = \{y_1(t), \ldots, y_N(t)\}$. This sequence of signals have been obtained as explained in Section 2.1.2, in a zone close to the impacted area. In order to reduce the computational cost, only a few signals have been chosen, representative of different levels of damage.

On the other hand, the quality of the fit between the modeled signals (which depend upon the model parameters $\boldsymbol{\theta}$, within a certain model-class $\mathcal{M}$) and the data $\mathcal{D}$ has to be assessed in this work. This information is given in terms of probability by a function called *likelihood function* ($p(\mathcal{D}|\boldsymbol{\theta}, \mathcal{M}_j)$). Both *posterior* and *likelihood* functions can be connected thanks to the Bayes' Theorem as follows:

$$\underbrace{p(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M}_j)}_{\text{posterior}} = c^{-1} p(\mathcal{D}|\boldsymbol{\theta}, \mathcal{M}_j) \underbrace{p(\boldsymbol{\theta}|\mathcal{M}_j)}_{\text{prior}} \tag{2.3}$$

where $p(\boldsymbol{\theta}|\mathcal{M}_j)$ represents prior information, specified as a Log-normal PDF (with the nominal values of the chosen parameters as the mean and a percentage of them as the standard deviation, see Table 3.1), defined before starting Bayesian inverse problem. The normalizing constant c is defined so that $p(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M}_j)$ represents a valid PDF, that is:

$$\int_{\boldsymbol{\Theta}} p(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M}_j) d\boldsymbol{\theta} = c^{-1} \int_{\boldsymbol{\Theta}} p(\mathcal{D}|\boldsymbol{\theta}, \mathcal{M}_j) p(\boldsymbol{\theta}|\mathcal{M}_j) \mathrm{d}\boldsymbol{\theta} = 1 \tag{2.4}$$

Solving this equation in order to obtain constant c^{-1} is quite challenging. Those integrals have not an analytical solution, so numerical solutions have to be chosen in order to compute this constant. But this constant is not relevant in order to obtain the *posterior* PDF shape. In addition, we use Markov-Chain Monte Carlo (MCMC) methods to simulate Bayesian inverse problem and obtain samples of this posterior (Section 2.4.4). After taking these samples we can obtain the constant c^{-1} by computing the evidence (see Section 2.4.7).

2.4.2 Bayesian model-class

The UWP model has to be "embedded" stochatically [18] in order to carry on defining the formulation of the likelihood function. The UWP model is established by a deterministic relationship $h_z = h_z(\mathbf{u}, \theta) : \mathbb{R}^{N_i} \times \mathbb{R}^{N_m} \rightarrow \mathbb{R}$, between the model input $\mathbf{u} \in \mathbb{R}^{N_i}$, that represents model information presented in Section 2.2 (Young's modulus, attenuation, etc.), and the model output $x(t) \in \mathbb{R}$, which expresses the modeled signal, given a set of N_p uncertain model parameters $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subset \mathbb{R}^{N_p}$. To be related with the UWP model, measured signals $y(t)$ can be expressed as below:

$$\underbrace{y(t)}_{\text{measured signal}} = \underbrace{h_z(t; \mathbf{u}, \boldsymbol{\theta})}_{\text{model}} + e(t) \tag{2.5}$$

The error term $e(t)$ represents the distance of the model to the measured signal. We choose to model the error $e(t)$ as a Gaussian PDF, which is supported by the Principle of Maximun Information Entropy (PMIE) [29, 30]. Consequently, the probability model for the measured signal $y(t)$ can be also modeled as a Gaussian PDF. This PMIE supplies us a tool for giving a probability model for the model-error term such that it produces the largest uncertainty (largest Shannon entropy). Thus, the stochastic UWP model can be defined as a function of model parameters $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, as

$$p(y|\mathbf{u}, \boldsymbol{\theta}, \mathcal{M}_j) = c_1 \exp\left(-J(y, \mathbf{u}, \boldsymbol{\theta})\right) \tag{2.6}$$

where c_1 is a normalizing constant and $J(y(t), \mathbf{u}, \boldsymbol{\theta})$ is a misfit function that give us information about how well the modeled signal and the experimental one fit each other. This function is defined as,

$$J(y, \mathbf{u}, \boldsymbol{\theta}) = \frac{1}{2} \int_0^T \left(\frac{y(t) - h_z(t; \mathbf{u}, \boldsymbol{\theta})}{\sigma(t)} \right)^2 dt \tag{2.7}$$

Once the expression that gives the relationship between the modeled and experimental signals in terms of probability is defined, the likelihood function can be formulated

2.4.3 Formulation of the *likelihood function*

An indicator of the degree of fitting is needed. To this point, the likelihood function can give us a view of this fitting. It is defined for N ultrasonic signals as data D, in case of we wanted to average a set of ultrasonic signals, by the stochastic UWP model defined in Equation (2.6), under damage hypothesis defined by the model-class $\mathcal{M}$, as follows:

$$p(\mathcal{D}|\boldsymbol{\theta}, \mathcal{M}_j) = \prod_{i=1}^{N} p(y_i(t)|\mathbf{u}, \boldsymbol{\theta}, \mathcal{M}_j) \tag{2.8}$$

By substituting Equation (2.6) into Equation (2.8), the likelihood function can be finally expressed as:

$$p(\mathcal{D}|\boldsymbol{\theta}, \mathcal{M}_j) = c_1 \exp\left(-\sum_{i=1}^{N} J(y_i, \mathbf{u}, \boldsymbol{\theta}) \right) \tag{2.9}$$

where c_1 is a normalizing constant.

Up to this point, the likelihood function and the prior information have been described by their PDF's. We can compute the shape of the *posterior* PDF of the parameters (Equation (2.3)) just knowing the prior and likelihood probability in the absence of knowing the constant defined in Section 2.4.1. In order to generate samples and taking them when they fit well with the data, the MCMC Metropolis-Hasting algorithm is used.

2.4.4 Metropolis-Hastings algorithm for Bayesian model updating

MCMC methods in Bayesian model updating generate parameter samples, which are distributed according to the *target* posterior PDF $p(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M}_j)$. One of the advantages of these methods is that they can avoid the specification of the constant presented in Equation (2.3), reducing the computational cost in obtaining the value of the integrals. This constant will be obtained in the calculus of the evidence. In particular, the Metropolis-Hasting (M-H) [31–33] algorithm has been selected because of its versatility and implementation simplicity.

The M-H algorithm can be applied for any problem, not only for signal approach problem, but for others like fatigue life prediction in composites [20, 34]. This algorithm is built up with a stationary distribution, which is the *posterior* PDF. Its implementation simplicity is one of its strengths, only a test about which *candidate model parameter* $\boldsymbol{\theta}'$ taken from a *proposal distribution* $q(\boldsymbol{\theta}'|\boldsymbol{\theta})$ is best suited to the posterior PDF is needed. The M-H obtains the state of the chain at $\zeta + 1$, given the state at ζ, specified by $\theta^{(\zeta)}$. The candidate parameter $\boldsymbol{\theta}'$ is accepted (i.e. $\boldsymbol{\theta}^{(\zeta+1)} = \boldsymbol{\theta}'$) with probability $\min\{1, r\}$, and rejected ($\boldsymbol{\theta}^{(\zeta+1)} = \boldsymbol{\theta}^{(\zeta)}$) with the remaining probability $1 - \min\{1, r\}$, where:

$$r = \frac{p(\mathcal{D}|\boldsymbol{\theta}^{'}, \mathcal{M}_j)p(\boldsymbol{\theta}^{'}|\mathcal{M}_j)q(\boldsymbol{\theta}^{(\zeta)}|\boldsymbol{\theta}^{'})}{p(\mathcal{D}|\boldsymbol{\theta}^{(\zeta)}, \mathcal{M}_j)p(\boldsymbol{\theta}^{(\zeta)}|\mathcal{M}_j)q(\boldsymbol{\theta}^{'}|\boldsymbol{\theta}^{(\zeta)})} \tag{2.10}$$

The *proposal distribution* $q(\boldsymbol{\theta}^{'}|\boldsymbol{\theta})$ is obtained from a multi-dimensional normal distribution, with the previous parameter of the chain as the mean and the standard deviation specified by a certain percentage of the prior samples (in the *prior distribution*) so that the acceptance ratio r is between a range 0.2-0.4 [35]. The process goes until N_s samples have been generated. This stop criterion is given by the cumulative average convergence of the target PDF. When this value converges, this sampling process can be stopped and the next step, the evidence of the model-class can be evaluated.

2.4.5 Bayesian model-class assessment

Up to this point, everything has been related to one particular damage hypothesis of the system represented by the model-class $\mathcal{M}_j$, but a sequence of model-classes can be taken in order to obtain which one fits better with data $\mathcal{D}$. This set of model-classes can be represented as a manifold: $\mathfrak{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_j, \ldots, \mathcal{M}_{N_M}\}$. Each of these model-classes is considered to be a different damage hypothesis, with different number of uncertain parameters. The probabilistic approach of model-class selection is motivated by the fact that the model itself may not necessarily reproduce the observed system, but it is just an approximation [30, 36]. Probabilities of each model-class can be obtained by making use of the Bayes' Theorem at the model-class level as:

$$p(\mathcal{M}_j|\mathcal{D}, \mathfrak{M}) = \frac{p(\mathcal{D}|\mathcal{M}_j)p(\mathcal{M}_j|\mathfrak{M})}{\sum_{i=1}^{N_M} p(\mathcal{D}|\mathcal{M}_i)p(\mathcal{M}_i|\mathfrak{M})} \tag{2.11}$$

where $p(\mathcal{M}_j|\mathfrak{M})$ is the prior probability of each model-class, that expresses the initial modeler's judgement on the relative degree of belief on $\mathcal{M}_j \in \mathfrak{M}$.

This prior probability has been taken with the same value for each model-class, since there is no information about the probability of occurrence. The factor $p(\mathcal{D}|\mathcal{M}_j)$ is the *evidence* (or *marginal likelihood*) for the model-class $\mathcal{M}_j$, and represents how likely the experimental signals are reproduced if model-class $\mathcal{M}_j$ is adopted. Note that the evidence is equal to the normalizing constant in establishing the posterior PDF in Equation (2.3), so that it can be obtained as:

$$p(\mathcal{D}|\mathcal{M}_j) = \int_{\boldsymbol{\Theta}} p(\mathcal{D}|\boldsymbol{\theta}, \mathcal{M}_j) p(\boldsymbol{\theta}|\mathcal{M}_j) \mathrm{d}\boldsymbol{\theta} \tag{2.12}$$

The evaluation of the last multi-dimensional integral is nontrivial, and can only be solved with numerical methods, except for some cases where the Laplace's method of asymptotic approximation can be used [37]. In this work, a recent technique based on samples from the posterior is adopted to numerically solve this integral [28].

2.4.6 Information-theory approach to model-class selection

To compute the evidence of a model-class, it is interesting to calculate the degree of robustness that the actual model-class holds. Robustness can be defined as an agreement between *data-fit* and *complexity* of the signal and this can be separately obtained. Thus, we can avoid extremes like over-fitting and under-fitting. A common principle enunciated is that, if data are explained equally well by two models, then the “simpler” one should be preferred (often referred to as Ockham's razor [30]). To tackle with this problem, Muto and Beck [38] proposed an information-theoretic interpretation of the evidence for a model-class, as follows:

$$\begin{aligned}\log p(\mathcal{D}|\mathcal{M}_j) &= \int_{\boldsymbol{\Theta}} [\log p(\mathcal{D}|\boldsymbol{\theta}, \mathcal{M}_j)]\, p(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M}_j)\boldsymbol{\theta} - \\ &- \int_{\boldsymbol{\Theta}} \left[\log \frac{p(\boldsymbol{\theta}\mathcal{D}, \mathcal{M}_j)}{p(\boldsymbol{\theta}|\mathcal{M}_j)}\right] p(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M}_j)\boldsymbol{\theta} = \\ &= \mathbb{E}[\log p(\mathcal{D}|\boldsymbol{\theta}, \mathcal{M}_j)] - \mathbb{E}\left[\log \frac{p(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M}_j)}{p(\boldsymbol{\theta}|\mathcal{M}_j)}\right]\end{aligned} \tag{2.13}$$

where $\mathbb{E}$ is the expectation with respect to the posterior PDF $p(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M}_j)$.

The first term of Equation (2.13) is a measure of the average goodness of fit (AGF) of the model-class $\mathcal{M}_j$ to the data $\mathcal{D}$. The second term is the *relative entropy* between the posterior and the prior PDFs, which measures the 'differenc", in logarithmic terms, between those PDFs [39]. This term determines the expected information gained (EIG) of the model-class $\mathcal{M}_j$ from the data and it is, by definition, always non-negative. The measure of data-fit of the model-class, specified before, corresponds to the AGF term in the Equation (2.13), whereas the complexity term is associated to the EIG. This equation provides us a quantitative value for the Principle of Model Parsimony or Ockham's razor [30]. Thus, the model-class that better fits the data and explains them in terms of information gained of data can be obtained.

2.4.7 Computation of the evidence for a model-class

Model-class posterior PDFs have to be obtained in order to choose which one fits better the data $\mathcal{D}$. This posterior probability will be calculated from the evidence of each model-class, and along with analysis of EIG and AGF terms, the best-suited model-class could be selected from a set of them. The computation of the evidence cannot be trivial, since this is a multi-dimensional problem. The Equation (2.12) could be solved with Monte Carlo methods, but computational cost would not be acceptable. To face this problem, Cheung and Beck [28] proposed a method based on samples from the posterior, and it has been adopted in this work. This technique analytically approximates the posterior PDF based on the samples obtained by the M-H algorithm. This approximation is used to obtain the evidence as:

$$\log p\left(\mathcal{D}|\mathcal{M}_j\right) = \log p\left(\mathcal{D}|\boldsymbol{\theta}, \mathcal{M}_j\right) + \log p\left(\boldsymbol{\theta}|\mathcal{M}_j\right) - \log p\left(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M}_j\right) \quad (2.14)$$

This equation is obtained by taking logarithms of Bayes' Theorem presented in Equation (2.3). All terms in Equation (2.14) can be analytically obtained, except the posterior PDF $p\left(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M}_j\right)$, whose information is based on samples. For more information about the algorithm used to obtain the evidence, see Section 2.5.1.

2.5 Algorithm description

To face this problem, with such huge set of possible parameters and model-classes, a blind damage parametrization algorithm is proposed. This algorithm goes through every layer and interface of the UWP model, establishing a model-class in each one. The Metropolis-Hasting algorithm, as depicted in Section 2.4.4, samples the space of parameters defined for each model-class, by comparing the ultrasonic signal generated with the forward problem with the experimental signal. When appropriated values are found, the target PDF tends to converge so the evidence calculation can be done. Once the algorithm goes trough all layers and interfaces a set of model-classes is obtained, so that the one which has higher posterior probability, and therefore fits better with the data, is selected.

The second stage of the search of model-classes begins after obtaining the first model-class (fixing the position of the layer/interface found in the first stage). In this second seeking, the algorithm tries again to assess model-classes in each layer and interface but with a difference. Indeed, in this second stage, there are two uncertain parameters, the fixed position from the last seeking (with the updated nominal value as the mean of the selected samples) plus the current position. This process goes until the modeled signal of the stage N fits worse the previous signal from the stage $N-1$.

2.5.1 Details of implementation

Figure 2.5 schematically depicts how the algorithm works. Firstly, decisions such as which variables select for layer/interfaces or to adjust factors for the M-H test have to be adopted before starting the algorithm computation. Secondly, the *prior* PDF of the chosen parameters is defined, sampling the space with a certain standard deviation. After this sampling, the *likelihood function* and the target *posterior* PDF are computed.

The test M-H is done bellow (Figure 2.7). The Markov chain engine provides samples for the target PDF. These samples must be evaluated into the likelihood function in order to know the degree of fitting for the sample given the data $\mathcal{D}$. The acceptance has a probability $\min\{1, r\}$ (see Equation (2.10)). This acceptance rate and the cumulative target PDF are evaluated in order to know if the M-H factors are appropriate. When the cumulative target PDF tends to converge into a single value this sub-algorithm can be stopped. Selected samples, which are chosen between a range defined by the burn-in period[4] and the limit[5] of the target PDF, are stored in a variable which contains the information used later in the evidence calculation.

In addition, the evidence is needed in order to make the Bayesian model-class assessment. The method based on samples from the posterior PDF

[4] This is the point of the cumulative *target* PDF since which, the tendency start being convergent.

[5] End of the cumulative *target* PDF once it has converged.

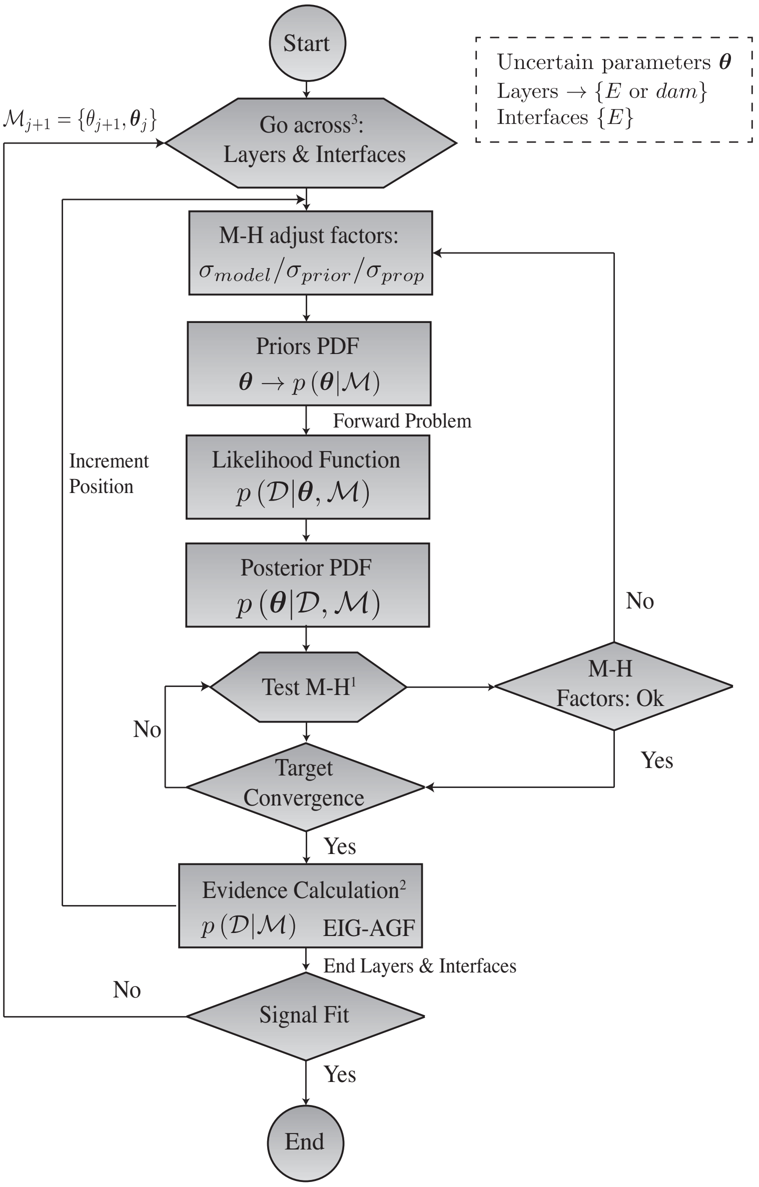

Start
Uncertain parameters $\boldsymbol{\theta}$
Layers $\rightarrow \{E$ or $dam\}$
Interfaces $\{E\}$
$\mathcal{M}_{j+1} = \{\theta_{j+1}, \boldsymbol{\theta}_j\}$
Go across[3]:
Layers & Interfaces
M-H adjust factors:
$\sigma_{model}/\sigma_{prior}/\sigma_{prop}$
Priors PDF
$\boldsymbol{\theta} \rightarrow p\left(\boldsymbol{\theta}|\mathcal{M}\right)$
Forward Problem
Increment
Position
Likelihood Function
$p\left(\mathcal{D}|\boldsymbol{\theta},\mathcal{M}\right)$
Posterior PDF
$p\left(\boldsymbol{\theta}|\mathcal{D},\mathcal{M}\right)$
No
Test M-H[1]
M-H
Factors: Ok
No
Yes
Target
Convergence
Yes
Evidence Calculation[2]
$p\left(\mathcal{D}|\mathcal{M}\right)$ EIG-AGF
End Layers & Interfaces
No
Signal Fit
Yes
End

Figure 2.5 Diagram of implementation of blind damage parametrization algorithm.(1) Test Metropolis-Hasting is depicted in Figure 2.7. (2) Evidence computation is described below.(3) Graphic explanation is depicted in Figure 2.6.

proposed by Cheung and Beck [28] is described bellow. Firstly, N_1 samples of the posterior PDF obtained in the M-H zone are selected in the range specified before and assessed in the posterior PDF. Secondly the Log-likelihood or AFG term in Equation (2.13) of the selected parameters is obtained, it depends on the values of the target and the prior PDF in the selected parameter as:

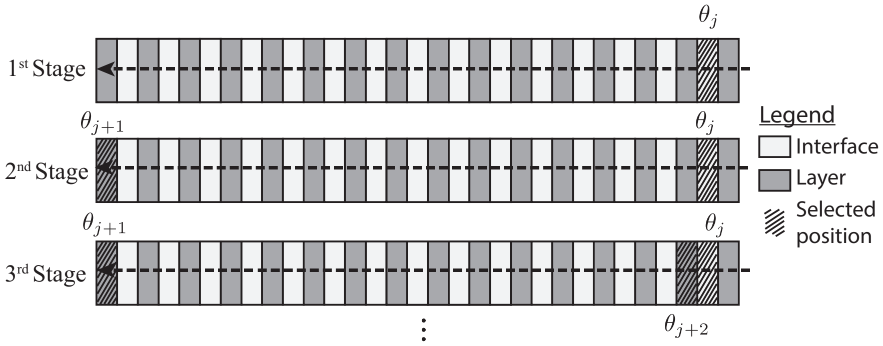


Figure 2.6 Diagram of stages. The algorithm goes through all layers and interfaces, choosing the most plausible position in each stage. Black arrows represent the direction of the tour.

$$\log p\left(\mathcal{D}|\boldsymbol{\theta},\mathcal{M}_j\right) = \frac{1}{N_1}\sum_{i=1}^{N_1}\frac{\log p\left(\boldsymbol{\theta}_i^{(k)}|\mathcal{D},\mathcal{M}_j\right)}{\log p\left(\boldsymbol{\theta}_i^{(k)}|\mathcal{M}_j\right)} \tag{2.15}$$

where $\boldsymbol{\theta}_i^{(k)}$ is the ith parameter in the set of selected ones. Two terms for obtaining the analytical approximation of the posterior have to be evaluated: the evaluation of the proposal, centred in the selected parameters and evaluating in the central parameter[6] $(\boldsymbol{\theta}_i^{(m)})$; $\epsilon^{(k)} = q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)})$ and the evaluation of the ratio between central parameters of the target and selected parameters $\tau^{(k)} = r(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)})$.

The analytical approximation of the posterior is carried out once the AFG term is calculated. The ratio between the target of proposed values and the target of the central value has to be addressed $\rho^{(m)} = r(\boldsymbol{\theta}^{(m)}|\boldsymbol{\theta})$. Once $\rho^{(m)}$, $\tau^{(k)}$ and $\epsilon^{(k)}$ are obtained, the analytical approximation of the posterior can be represented as:

$$p\left(\boldsymbol{\theta}|\mathcal{D},\mathcal{M}_j\right) \approx \frac{\frac{1}{N_1}\sum_{k=1}^{N_1}\tau^{(k)}\epsilon^{(k)}}{\frac{1}{N_2}\sum_{m=1}^{N_1}\rho^{(m)}} \tag{2.16}$$

The evaluation of the Log-evidence can be reached with the Equation (2.14).

[6] Central parameters are picked up from a N_2 vector of the central part of the selected parameters, organized by the number of proposal values given in the M-H part of the algorithm.

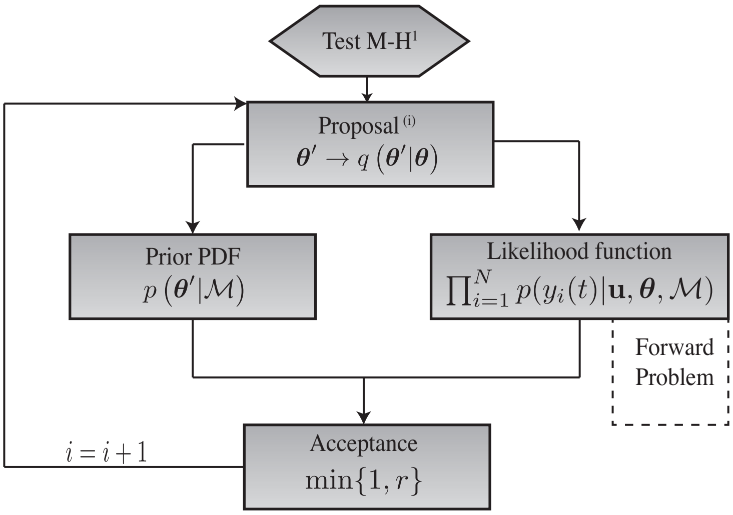

Test M-H[1]
Proposal (i)
$\boldsymbol{\theta}' \rightarrow q\left(\boldsymbol{\theta}'|\boldsymbol{\theta}\right)$
Prior PDF
$p\left(\boldsymbol{\theta}'|\mathcal{M}\right)$
Likelihood function
$\prod_{i=1}^{N} p(y_i(t)|\mathbf{u}, \boldsymbol{\theta}, \mathcal{M})$
Forward Problem
Acceptance
$\min\{1, r\}$
$i = i + 1$

Figure 2.7 Diagram of implementation of Metropolis-Hasting test algorithm.This is a explanation of the Figure 2.5 in its part *Test M-H*.

On the other hand, the term of the Log-evidence named as EIG in Equation (2.13) can be addressed by extracting the Log-evidence to the AGF term.

Moreover, all model-classes (for layers and interfaces for N parameters) are assessed in order to obtain that which fits better with data. This process stops when the modeled signal is sufficiently well fitted with the experimental one and when it does not improve the matching of the modeled signal from the previous stage.

Chapter 3

Results

This chapter describes the results obtained from the blind parametrization algorithm. Firstly, Section 3.1 presents the relevant information used in the algorithm for signal interpretation. Then, Section 3.2 focuses on the algorithm evaluation by making use of synthetically damaged signals described in Section 2.3. Finally, some experimental signals obtained from the damaged CFRP plate described in Section 2.1 are analyzed in Section 3.3, in order to highlight the potential of our proposal.

3.1 Algorithm specifications

The blind parametrization algorithm has been calibrated for both synthetic and experimental signals. The prior information on damage parameters is listed in Table 3.1. The prior PDF selected in this work is a lognormal (LN) distribution centred in the nominal value of each parameter, since the mechanical values of these parameters are always non-negative, and with a standard deviation defined by a percentage of variation of its nominal value. The coefficients of variation (C.O.V.) for each parameter are chosen with a relatively large dispersion since the degree of changes due to damage is *a priori* unknown. The subindex $i = 1, \ldots, 31$ is referred to each layer and interface as depicted in Figure 3.1 (be aware that the 2 water layers are not considered here, since they remain unchanged).

Localization	Parameter	Nominal value	Units	C.O.V. (%)	Prior PDF
Layers	E_i	11.1616	GPa	40	LN
	α_i	293.02	Np/m	60	LN
Interfaces	E_i	5.2728	GPa	40	LN

Table 3.1 *Prior information of the parameters used in calculations.*

Model error parameters (standard deviation in the misfit function, see Section 2.4.2) are fixed with a deterministic value for each kind of damage. The smaller one was selected for the signal from a lowly damaged area, while increasing its value for the signals situated in a highly damaged area. The C.O.V. of this model error parameter is selected to be 5% for the signal from a lowly damaged area, 8% for the signal from a moderately damaged area and 10% for the signal from a highly damaged area.

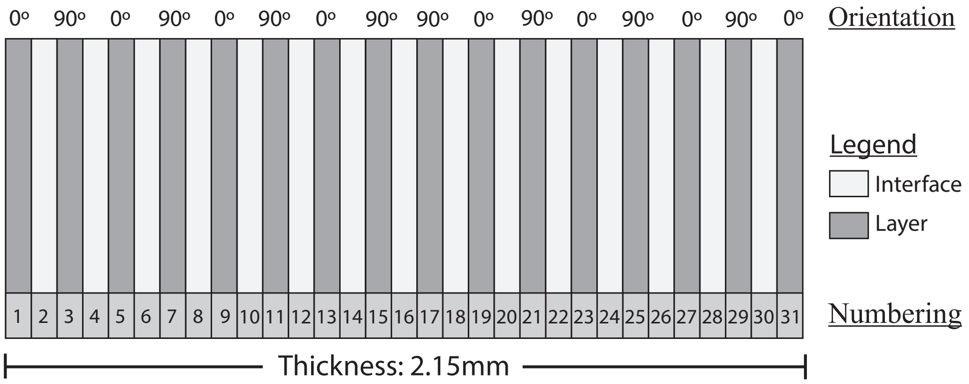


Figure 3.1 CFRP numbering scheme for parameter subscripts.

The burn-in period (see Section 2.5.1) is automatically selected in each model-class assessment. Concretely, once the cumulative posterior probability tends to converge, the burn-in period is selected around this point.

3.2 Evaluation of the algorithm

The proposed algorithm is first evaluated by making use of synthetic signals. These signals were directly generated with the UWP model (see Section 2.2) by inserting synthetic damage in the physical parameters. Damage was induced in some layers and interfaces by varying the nominal value of the selected parameter, that is by reducing the value of the Young modulus E.

3.2.1 Test 1

The algorithm procedure used for obtaining the damage parametrization is depicted step by step for this first test using a synthetic signal. Test number 1 consists of a synthetic signal damaged in the first layer, and the first and last interfaces. The nominal values of the corresponding Young moduli were reduced as indicated in Table 3.2.

Parameter	$E_{i=1}$ (GPa)	$E_{i=2}$ (GPa)	$E_{i=30}$ (GPa)
Nominal value	11.1616	5.2728	5.2728
Synthetic damage	8	4	4

Table 3.2 *Nominal values and the synthetic damage induced in some layers/interfaces.*

The ultrasonic signal modeled with those altered nominal values is now chosen as the *experimental* one to evaluate our blind parametrization algorithm. The goal of this test is twofold: (1) To assess the level of precision of the algorithm bearing in mind that IP's may suffer from unicity problem, that is two model parametrizations could give the same solution; (2) to evaluate the robustness to noise of the algorithm by adding three different levels of WGN to the signal, which respectively correspond to a SNR of 25, 20 and 15 dB.

The model-class selection is briefly described for the noise-free synthetic signal in order to illustrate how the algorithm searches for the best-suited damage parameters. This search is done through each parameter of the model, which has been previously designated to be uncertain. Layer by layer and interface by interface, the Log-evidence and therefore the posterior probability of each model class are assessed, so that the best one is selected in each stage and its position fixed for the new search. The posterior probability of each class is presented in Figure 3.2 for the first stage. As can be observed, the first-model class $\mathcal{M}_1$ reaches the highest probability.

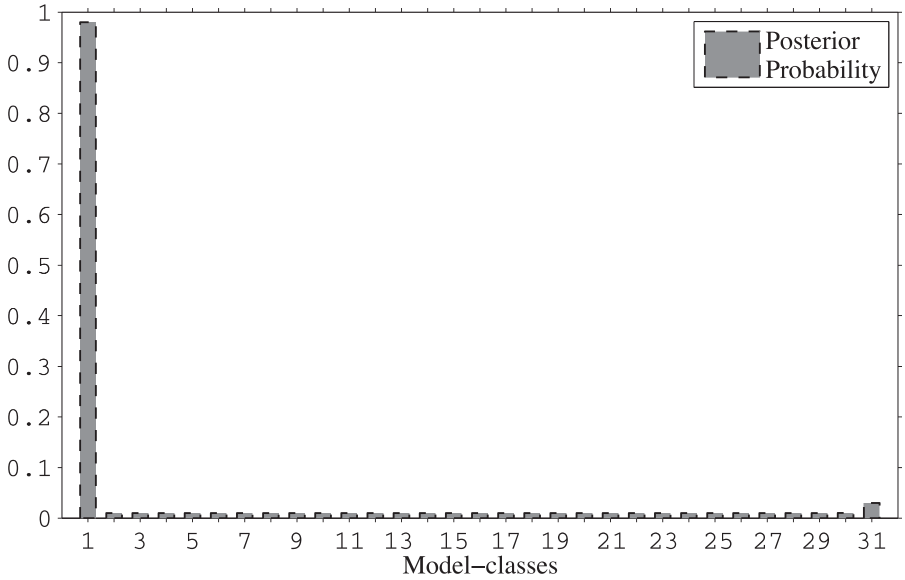


Figure 3.2 Posterior probability of each class for the first stage. The best-suited damage parameter is the model-class $\mathcal{M}_1$.

The two next stages have model-classes that are less differentiable. The three model-classes along with their posterior probabilities are summarized in the Table 3.3. Posterior probability is more identificable in the first stage, whereas it is less differentiable in the two next stages, since the most important improvement in the signal fitting is due to first parameter (that is the parameter in the first layer, which absorbs most of the damage information in contrast to the interfaces). As can be observed, the synthetically damaged parameters (see Table 3.2) have been chosen in a proper way (the mean and standard deviation of the chosen parameters are depicted in Table

3.4).

Iteration	1^{st} Stage	2^{nd} Stage	3^{rd} Stage
Model-class	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_{30}$
Posterior probability	0.99657	0.39746	0.074405

Table 3.3 *Model-classes selected in each stage of the algorithm, along with their respective posterior probabilities.*

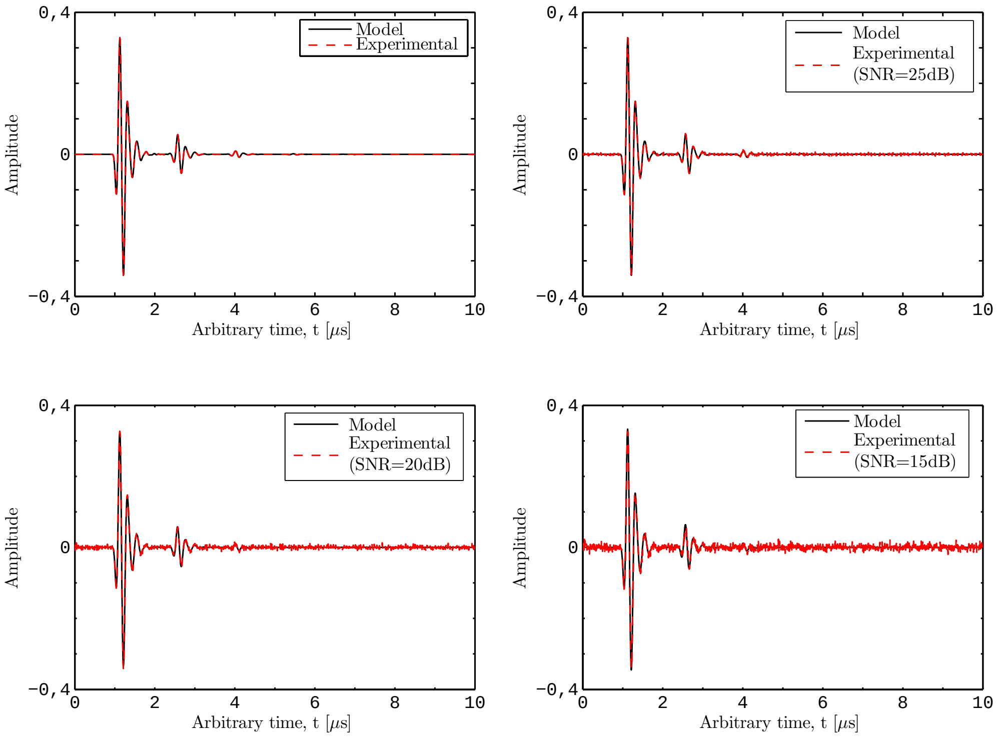


Figure 3.3 Comparison between synthetic signal without noise (upper-left plot), with SNR=25 dB (upper-right plot), with SNR=20 dB (lower-left plot) and SNR=15 dB (lower-right plot) and the mean of modeled signals given by the samples obtained from the blind parametrization algorithm.

In addition, Figure 3.3 shows how well-fitted are the experimental and the modeled signals for the different levels of noise. In the upper-left plot, each point of the *experimental* measurement is reliably reproduced by our blind damage parametrization. In the upper-right plot, the matching is remarkable for this low level of noise (SNR = 25 dB) and further supports the performance of the proposed algorithm. Similar results have been obtained in the lower-left plot (SNR = 20 dB). As can be observed in the lower-right plot, the synthetic signal with a SNR = 15 dB is also well-fitted with the modeled signal. The slight signal mismatch is due to the high noise level (i.e. the third echo is totally masked by noise) and the fact that several model-classes are possible candidates to match the experimental signal. Despite the good matching the selection of parameters was not very accurate for this last reconstruction (see Table 3.4).

Furthermore, Figure 3.4 depicts the scatter plot of the reconstructed parameters in the case of the synthetic signal without noise. As can be seen, the reconstructed values fit well those specified in Table 3.2. It can be observed that the first parameter is more accurate than the others, which thus supports that the parameter in the first stage is more identifiable.

Damaged parameter	$E_{i=1}$ GPa	$E_{i=2}$ GPa	$E_{i=30}$ GPa
Assigned values	8	4	4
Synthetic w/o noise	7.99 ± 0.12	4.26 ± 0.73	4.04 ± 0.46
Synthetic SNR = 25 dB	7.99 ± 0.12	4.26 ± 0.72	3.96 ± 0.45
Synthetic SNR = 20 dB	7.55 ± 0.37	4.28 ± 0.64	3.62 ± 0.72
Synthetic SNR = 15 dB	8.64 ± 1.52	4.32 ± 1.08	–

Table 3.4 *Parameter reconstruction for each different synthetic signal in test 1. Empty cells mean that the parameter delivered by the algorithm was in an erroneous position.*

Table 3.4 collects the mean and standard deviation of the chosen parameters in the last stage of the algorithm. These were computed as follows,

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\theta}_i \quad \text{and} \quad \boldsymbol{\sigma} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{\theta}_i - \boldsymbol{\mu})^2} \tag{3.1}$$

For the three first cases, the latter are in good agreement with the synthetically damaged parameters provided in Table 3.2. Nevertheless, in the fourth synthetic signal (SNR = 15 dB) the algorithm did not identify the last interface. The third parameter chosen by the algorithm in the last stage

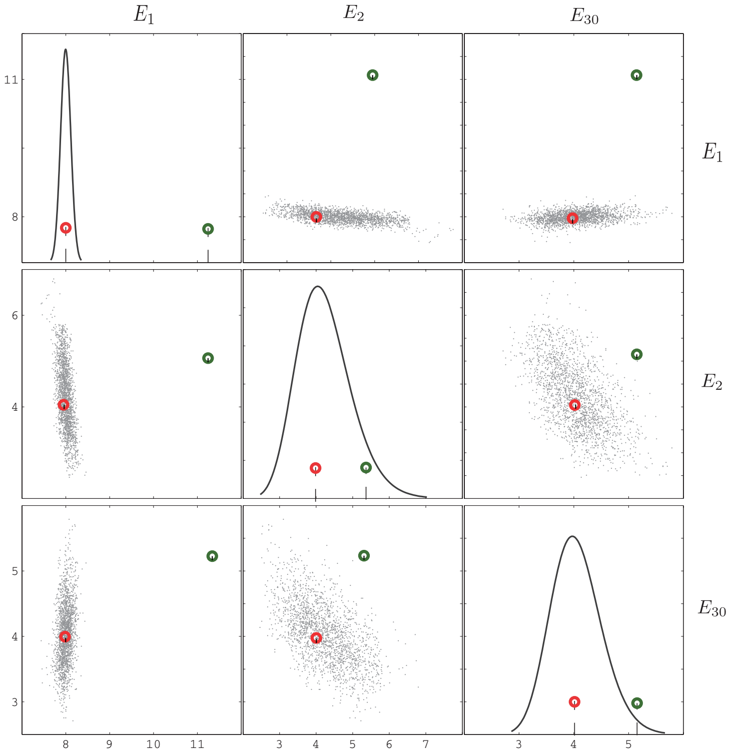


Figure 3.4 Plots of 1e+5 posterior samples when updating model class $\mathcal{M}_{30}$ in the third stage with synthetic signal without noise of test 1 data $\mathcal{D}$. On the diagonal, lognormal fit of the samples shows the marginal PDF of each parameter. Green circles represent the undamaged parameters in each scatter plot, on the diagonal, this symbol indicates the undamaged parameter for the ordinate axis. Red circles represent the synthetically damaged parameters.

was the last layer. The algorithm lacks in precision due to the high level of noise, which, in this case, altered the convergence.

3.2.2 Test 2

Test number 2 consists of a synthetic signal damaged in two layers, the tenth and the fifteenth one, and three interfaces, the first, last and ninth ones, as presented in Table 3.5.

The ultrasonic signal modeled with these new values of the parameters is chosen as the experimental one in test number 2. Again, the robustness to noise of the algorithm is evaluated by adding three different levels of WGN as in Test 1 (see Section 3.2.1).

Parameter	$E_{i=19}$ (GPa)	$E_{i=29}$ (GPa)	$E_{i=2}$ (GPa)	$E_{i=18}$ (GPa)	$E_{i=30}$ (GPa)
Nominal value	11.1616	11.1616	5.2728	5.2728	5.2728
Synthetic damage	6	9	3	3.5	3.7

Table 3.5 *Nominal values and the synthetic damage induced in some layers/interfaces.*

The posterior probabilities of all the model-classes selected by the algorithm given the experimental signal without noise are shown in the Table 3.6.

Iteration	1^{st} Stage	2^{nd} Stage	3^{rd} Stage	4^{nd} Stage	5^{rd} Stage
Model-class	$\mathcal{M}_{19}$	$\mathcal{M}_{29}$	$\mathcal{M}_{2}$	$\mathcal{M}_{30}$	$\mathcal{M}_{18}$
Posterior probability	0.9999	0.2018	0.1062	0.05614	0.05751

Table 3.6 *Model-classes selected in each stage of the algorithm, as well as their respective posterior probabilities for the synthetic signal without noise.*

As in Test 1 (Section 3.2.1), it can be observed that high probability relies on the first model-class of the algorithm. The next stages have a lower difference between their respective model-classes posterior probabilities, since the most important improvement in the signal fitting is again due to first parameter. As can be observed, the synthetically damaged parameters (see Table 3.5) have been chosen in a proper way for all the noise levels.

In addition, Figure 3.5 shows the degree of matching between the experimental and the modeled signals. In the upper-left plot, the modeled signal by the algorithm shows high standard of fit with the experimental one. In the upper-right plot, the matching is remarkable for this low level of noise (SNR = 25 dB), and even with the higher levels of noise showed in the lower plots, the signal fitting between experimental and modeled signals are notable.

Moreover, Figure 3.6 depicts the scatter plot of the reconstructed parameters in the case of the synthetic signal without noise. The reconstructed

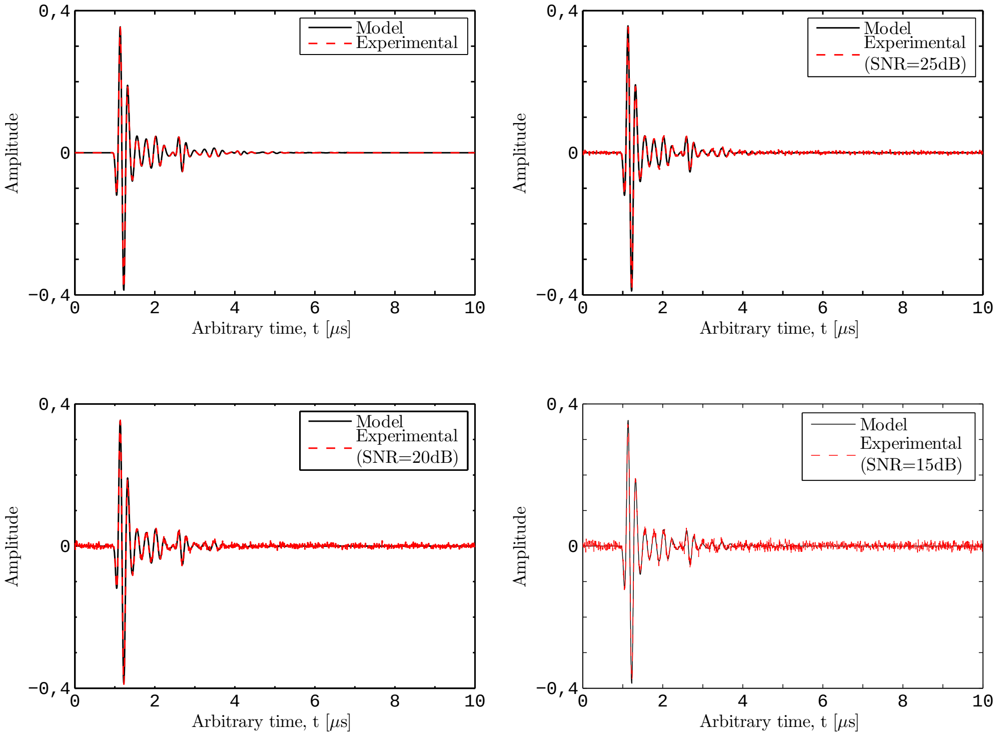


Figure 3.5 Comparison between synthetic signal without noise (upper-left plot), with SNR=25 dB (upper-right plot), with SNR=20 dB (lower-left plot) and SNR=15 dB (lower-right plot) and averaged modeled signal given by the samples obtained from the blind parametrization algorithm.

values are close to those which were specified in Table 3.5. It can be observed that the first and second parameters are more accurate than the others. This supports that parameters in the first stages (i.e. the layers) are better identificated. The spread of the values increases when considering the interfaces (the three last columns and rows).

Damaged parameter	$E_{i=19}$	$E_{i=29}$	$E_{i=2}$	$E_{i=18}$	$E_{i=30}$
	GPa	GPa	GPa	GPa	GPa
Assigned values	6	9	3	3.5	3.7
Synthetic w/o noise	6.05 ± 0.26	9.22 ± 0.45	4.36 ± 0.81	4.28 ± 0.93	3.69 ± 0.67
Synthetic SNR = 25 dB	6.33 ± 0.48	9.40 ± 0.61	4.71 ± 1.19	4.73 ± 1.20	4.16 ± 0.86
Synthetic SNR = 20 dB	6.00 ± 0.38	9.10 ± 0.66	4.29 ± 0.62	3.25 ± 0.77	3.68 ± 0.71
Synthetic SNR = 15 dB	6.06 ± 0.26	9.14 ± 0.45	4.39 ± 0.75	4.40 ± 0.88	3.75 ± 0.68

Table 3.7 *Parameter reconstruction for each different synthetic signal in Test 2.*

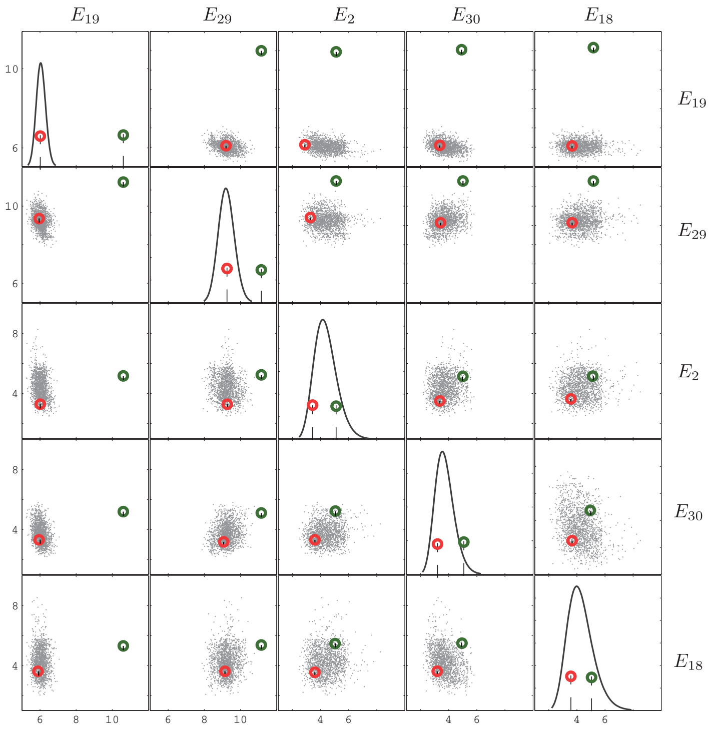


Figure 3.6 Plots of 8e+4 posterior samples when updating model class $\mathcal{M}_{30}$ in the third stage with synthetic signal without noise of test 2 data $\mathcal{D}$. On the diagonal, lognormal fit of the samples shows the marginal PDF of each parameter. Green circles represent the undamaged parameters in each scatter plot, on the diagonal, this symbol indicates the undamaged parameter for the ordinate axis. Red circles represent the synthetically damaged parameters.

The mean and standard deviation of the chosen parameters in the last stage of the algorithm are shown in Table 3.7. It can be seen that the first and second parameters, which stand for the Young's modulus in the layers, fit better with the synthetic ones (6 GPa and 9 GPa). The presence of noise in the signal does not distort the mean of the two first parameters, but causes a larger dispersion for the three last parameters that correspond to the Young moduli of the interfaces (3 GPa, 3.5 GPa and 3.7 GPa). For this test, despite the high number of model parameters, the algorithm provides a reliable guess for the reconstruction of all the parameters, independently of the considered noise level.

3.2.3 Test 3

Test number 3 consists of a synthetic signal damaged in three layers, the first, eighth and the ninth ones, and the last interface, summarized in Table 3.8.

Parameter	$E_{i=1}$ (GPa)	$E_{i=15}$ (GPa)	$E_{i=17}$ (GPa)	$E_{i=30}$ (GPa)
Nominal value	11.1616	11.1616	11.1616	5.2728
Synthetic damage	6	8	8	3

Table 3.8 *Nominal values and the synthetic damage induced in some layers/interfaces.*

These new values of the parameters are introduced into the forward problem, resulting in the ultrasonic signal, which is chosen as the experimental one in this third assessment of the algorithm. Again, the robustness to noise of the algorithm is evaluated by adding three different levels of WGN as in the two previous tests.

Furthermore, Figure 3.7 shows the signal reconstruction from the selected parameters in the algorithm. It can be seen that in the upper-left and upper-right plots, which correspond to the signal without noise and with a SNR of 25 dB respectively, the signal fitting is remarkable. Despite the the level of noise, the blind algorithm correctly chose the parameters in these two first cases, but this does unfortunately not occur with the higher levels of noise. The lower plots show how the modeled signals are reasonably well fitted, but they differ from the experimental one, especially in the case of the highest noise level. The mismatch can be explained due to a random selection of the uncertain parameters by the algorithm, because some damage information may be drown in noise.

The mean and standard deviation of these uncertain parameters selected by the algorithm are shown in Table 3.9. For the signals without noise and with the lower level of noise (SNR = 25 dB) the three first parameters

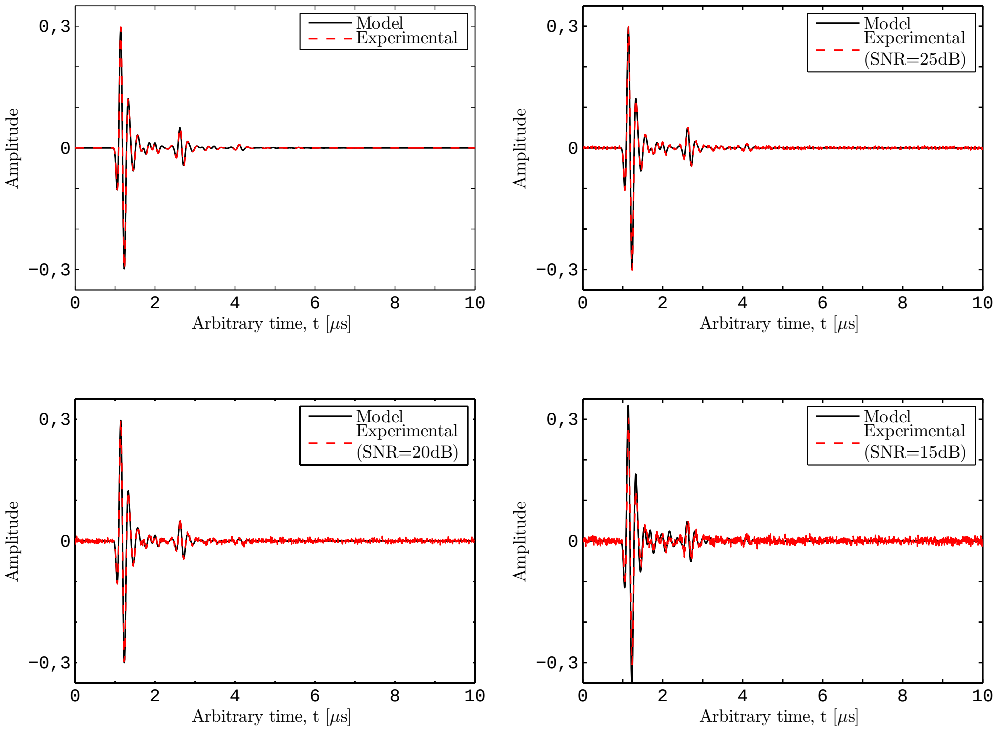


Figure 3.7 Comparison between synthetic signal without noise (upper-left plot), with SNR=25 dB (upper-right plot), with SNR=20 dB (lower-left plot) and SNR=15 dB (lower-right plot) and averaged modeled signal given by the samples obtained from the blind parametrization algorithm.

Damaged parameter	$E_{i=1}$ GPa	$E_{i=15}$ GPa	$E_{i=17}$ GPa	$E_{i=30}$ GPa
Assigned values	6	8	8	3
Synthetic w/o noise	5.95 ± 0.23	8.23 ± 0.56	8.09 ± 0.51	4.7157 ± 1.33
Synthetic SNR = 25 dB	5.98 ± 0.22	8.21 ± 0.51	8.07 ± 0.58	4.45 ± 1.18
Synthetic SNR = 20 dB	6.11 ± 0.35	8.10 ± 0.97	7.64 ± 0.92	–
Synthetic SNR = 15 dB	9.59 ± 0.90	–	6.44 ± 0.33	–

Table 3.9 *Parameter reconstruction for each different synthetic signal in test 3. Empty cells mean that the parameter delivered in the algorithm was in an erroneous position.*

have been chosen with more accuracy than the last one, that is the Young's modulus in the last interface. Nevertheless, with the higher levels of noise, the algorithm failed in the parameter selection. This is due to the high level of noise, which limits the algorithm performance. It must also be noted that for this large induced synthetical damage, the echoes are drastically reduced in amplitude, and nearly completely drown in noise.

Moreover, error in the mean of the reconstructed parameters for tests 1 and 3 are depicted in Figure 3.8. The relationship between measurement noise and reconstruction noise should be linear in a log-log scale if the forward and IP's were linear. The obtained results serve as a further illustration of the nonlinearity of the IP, which is a cause of its ill-posedness in the sense of Hadamard [40], and the physical complexity of the problem.

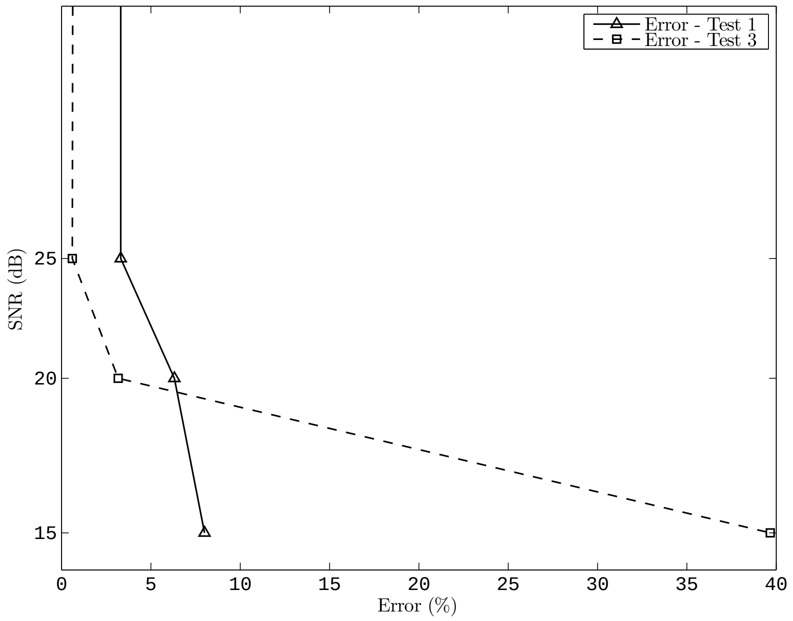


Figure 3.8 Relationship between the error percentage in the parameter reconstruction and the level of noise in terms of SNR.

3.3 Experimental evaluation

Experimental signals measured on the damaged CFRP plate, as explained in Section 2.1, are now introduced as data in our blind parametrization algorithm. Three different experimental signals were chosen, in order to represent three levels of damage with increasing complexity. The goal of this section is to evaluate if our algorithm is able to reconstruct signals obtained from real inspection conditions. Since damage in layers can be *a priori* explained with a decrease in the Young's modulus E (first configuration) or an increase in the attenuation coefficient α (second configuration), the experimental signals are assessed by comparing both computations in order to obtain the most reliable damage configuration. The Young's modulus is the selected parameter considered as uncertain for the interfaces in both cases.

3.3.1 Low damage

The first measurement was taken from a low-damaged area of the CFRP plate. Figure 3.9 shows both reconstructions for the chosen configuration in the layers with Young's modulus (left) and attenuation coefficient (right). A fairly good matching is obtained for both configurations. The reconstruction performed with the attenuation coefficients fits slightly better the experimental signal than that performed with the Young modulus.

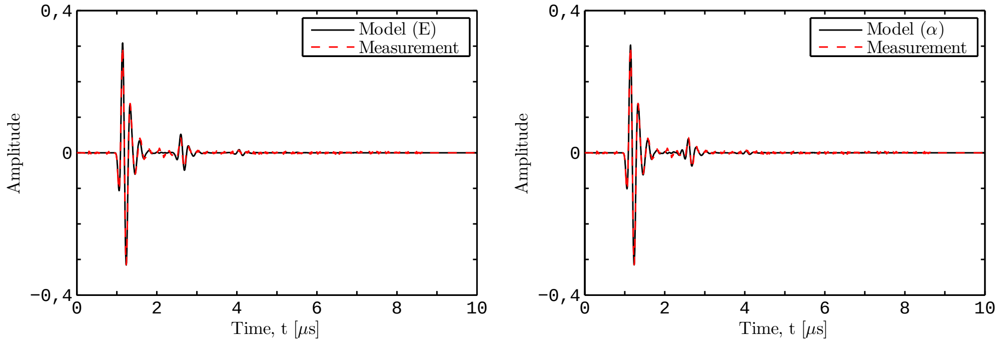


Figure 3.9 Comparison between experimental measurement and signal reconstruction with the first configuration on the left, and second configuration on the right.

Since both configurations are different, the parameter reconstruction has been addressed separately. The first reconstruction shows a decrease in the nominal values of the Young modulus for the external layers and interfaces (see Table 3.10). This behavior is consistent with the initial phases of impact damage, where matrix cracks are generated in the back-wall layers, followed by the growth of delaminations, and if the impact energy is sufficiently higher, these defects can also appear on the impacted layer of the CFRP plate [9].

	First configuration			Second configuration		
	Parameter	Mean±Std	Nominal	Parameter	Mean±Std	Nominal
Layers	E_1 (GPa)	6.30 ± 0.20	11.1616	α_7 (Np/m)	215.53 ± 248	293.02
	E_{29} (GPa)	9.95 ± 0.51	11.1616	α_{19} (Np/m)	1425.9 ± 315	293.02
Interfaces	E_2 (GPa)	4.54 ± 0.51	5.2728	E_2 (GPa)	1.93 ± 0.30	5.2728
	E_{30} (GPa)	4.14 ± 0.59	5.2728	E_{28} (GPa)	1.95 ± 0.11	5.2728
	–	–	–	E_6 (GPa)	2.58 ± 0.34	5.2728

Table 3.10 *Parameter reconstruction for both configurations.*

On the other hand, the second reconstruction shows a relatively different behavior than the first configuration. The mean values of the chosen parameters by the algorithm are shown in Table 3.10. As can be observed, the damage in the interfaces is chosen at the same locations, whereas the altered attenuation coefficients predict a damage in layers located deeper within the specimen. Despite the good reconstruction, this behavior seems not to be as realistic as the one obtained in the first configuration.

3.3.2 Moderate damage

The second measurement was taken from a moderate-damaged area of the CFRP plate. Figure 3.10 shows both reconstructions for the chosen configuration in the layers with Young's modulus (left) and attenuation coefficient (right). A relatively good matching is obtained for both configurations. As can be observed, the left plot fits better the second part of the wave, which comes after the wave-front. On the other hand, the right plot fits better the wave-front, whereas the matching of the echoes is relatively poor.

Since both configurations are different, the parameter reconstruction has been addressed separately again. The first reconstruction shows a decrease in the nominal values of the Young's modulus for the external layers as well

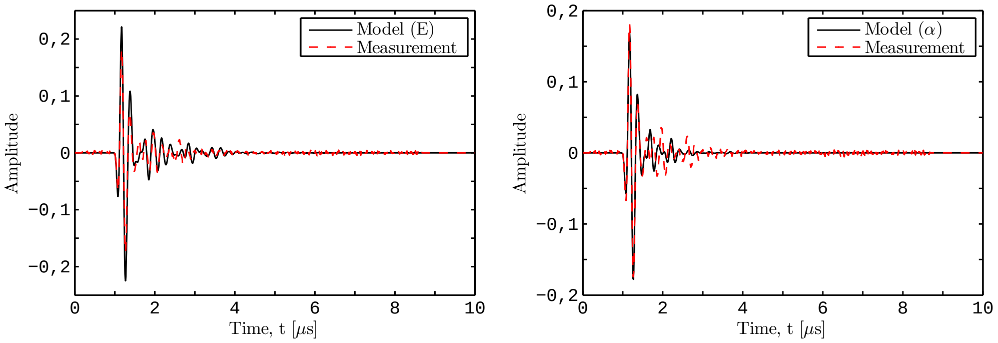


Figure 3.10 Comparison between experimental measurement and signal reconstruction with the first configuration on the left, and the second configuration on the right.

as in the layers situated in the middle, as summarized in Table 3.11. In addition, a decrease in the nominal value of the Young's modulus for one interface, which is stacked with a damaged layer, has been selected.

	First configuration			Second configuration		
	Parameter	Mean±Std	Nominal	Parameter	Mean±Std	Nominal
Layers	E_1 (GPa)	9.13 ± 1.05	11.1616	α_{13} (Np/m)	214.6 ± 232	293.02
	E_{31} (GPa)	5.96 ± 0.42	11.1616	α_3 (Np/m)	2536.7 ± 464	293.02
	E_{15} (GPa)	3.12 ± 0.83	11.1616	α_{21} (Np/m)	2420.7 ± 563	293.02
	E_{17} (GPa)	3.56 ± 0.34	11.1616	–	–	–
	E_5 (GPa)	4.56 ± 0.80	11.1616	–	–	–
Interfaces	E_4 (GPa)	0.88 ± 0.11	5.2728	E_{12} (GPa)	2.79 ± 0.99	5.2728
	–	–	–	E_{22} (GPa)	0.53 ± 0.05	5.2728

Table 3.11 *Parameter reconstruction for both configurations.*

On the other hand, the second reconstruction chooses a deeper damage pattern within the specimen. The mean values of the chosen parameters by the algorithm are shown in Table 3.11. As can be observed, the damage is placed in different layers/interfaces compared to first reconstruction. The damage in the interfaces is next to the damage in layers. The change in the the mean value is considerable.

3.3.3 Severe damage

The last experimental signal was chosen in a zone within the impacted area. Figure 3.11 shows both reconstructions as explained before. In the left plot, which represents signal reconstruction with the first reconstruction, the modeled signal fits worse the wave front than the modeled signal presented in the right plot. Even though, the reconstruction is quite poor for both cases. In the left plot, the algorithm tries to fit the echoes, but the matching is relatively bad. In contrast, in the right plot, the algorithm fits well the wave front, but totally ignores the echoes, probably due to the high noise level.

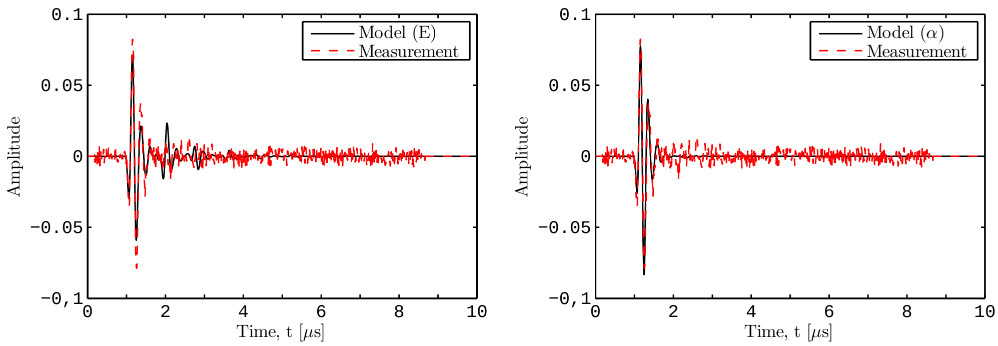


Figure 3.11 Comparison between experimental measurement and signal reconstruction with the first configuration on the left, and the second configuration on the right.

	First configuration			Second configuration		
	Parameter	Mean±Std	Nominal	Parameter	Mean±Std	Nominal
Layers	E_5 (GPa)	9.46 ± 1.07	11.1616	α_1 (Np/m)	265.28 ± 415	293.02
	E_{29} (GPa)	9.10 ± 1.12	11.1616	α_{25} (Np/m)	13181 ± 1028	293.02
	E_{27} (GPa)	7.10 ± 1.29	11.1616	–	–	–
	E_{19} (GPa)	7.02 ± 1.23	11.1616	–	–	–
	E_{25} (GPa)	6.05 ± 1.06	11.1616	–	–	–
Interfaces	E_{26} (GPa)	3.79 ± 0.80	5.2728	–	–	–
	E_{18} (GPa)	0.06 ± 0.01	5.2728	–	–	–

Table 3.12 *Parameter reconstruction for both configurations.*

The first reconstruction chooses a decrease in the nominal values of

Young's modulus in layers of the latter half of the CFRP plate, the opposite to the impacted face, as summarized in Table 3.12. It is observed that the damage is dispersed in the opposite layers to the impacted face.

On the other hand, the second reconstruction chooses a simpler pattern of damage since the damage is concentrate in one layer by a drastic increment in the attenuation coefficient compared to its nominal value. The mean values of the selected parameters as uncertain are presented in Table 3.12.

	First reconstruction					
	Low damage		Moderate damage		Severe damage	
Iteration	$\mathcal{M}_i$	Post. (%)	$\mathcal{M}_i$	Post. (%)	$\mathcal{M}_i$	Post. (%)
1^{st} stage	$\mathcal{M}_1$	99.9	$\mathcal{M}_1$	99.9	$\mathcal{M}_{26}$	47.14
2^{nd} stage	$\mathcal{M}_2$	9.37	$\mathcal{M}_{31}$	90.78	$\mathcal{M}_5$	5.20
3^{rd} stage	$\mathcal{M}_{29}$	6.75	$\mathcal{M}_4$	96.95	$\mathcal{M}_{29}$	4.63
4^{th} stage	$\mathcal{M}_{30}$	5.80	$\mathcal{M}_{15}$	18.03	$\mathcal{M}_{27}$	4.39
5^{th} stage	-	-	$\mathcal{M}_{17}$	47.02	$\mathcal{M}_{19}$	4.28
6^{th} stage	-	-	$\mathcal{M}_5$	5.81	$\mathcal{M}_{18}$	4.34
7^{th} stage	-	-	-	-	$\mathcal{M}_{25}$	4.70

	Second reconstruction					
	Low damage		Moderate damage		Severe damage	
Iteration	$\mathcal{M}_i$	Post. (%)	$\mathcal{M}_i$	Post. (%)	$\mathcal{M}_i$	Post. (%)
1^{st} stage	$\mathcal{M}_7$	12.82	$\mathcal{M}_{13}$	14.26	$\mathcal{M}_1$	7.09
2^{nd} stage	$\mathcal{M}_2$	6.32	$\mathcal{M}_{12}$	28.48	$\mathcal{M}_{25}$	3.60
3^{rd} stage	$\mathcal{M}_{28}$	10.03	$\mathcal{M}_3$	29.71	-	-
4^{th} stage	$\mathcal{M}_6$	21.99	$\mathcal{M}_{21}$	3.41	-	-
5^{th} stage	$\mathcal{M}_{19}$	4.52	$\mathcal{M}_{22}$	6.75	-	-

Table 3.13 *Model-class plausibilities for each parameter/stage for the two reconstructions in each level of damage (Experimental evaluation measurements).*

Finally, the model-class plausibilities of the selected parameters are presented in Table 3.13. The first and second configurations are presented together, in addition to the three different parametrizations for each experimental signal in order to give a global view of the plausibilities. As can be observed, the model-class plausibilities for the first configuration are higher than the second one. In addition, the model-class plausibilities placed in the layers, in the first stages for the first configuration are also higher than these ones for the second configuration. It can be also noted that the more number of the algorithm stages, the less plausibility of the selected model-class.

Chapter 4

Discussion and conclusions

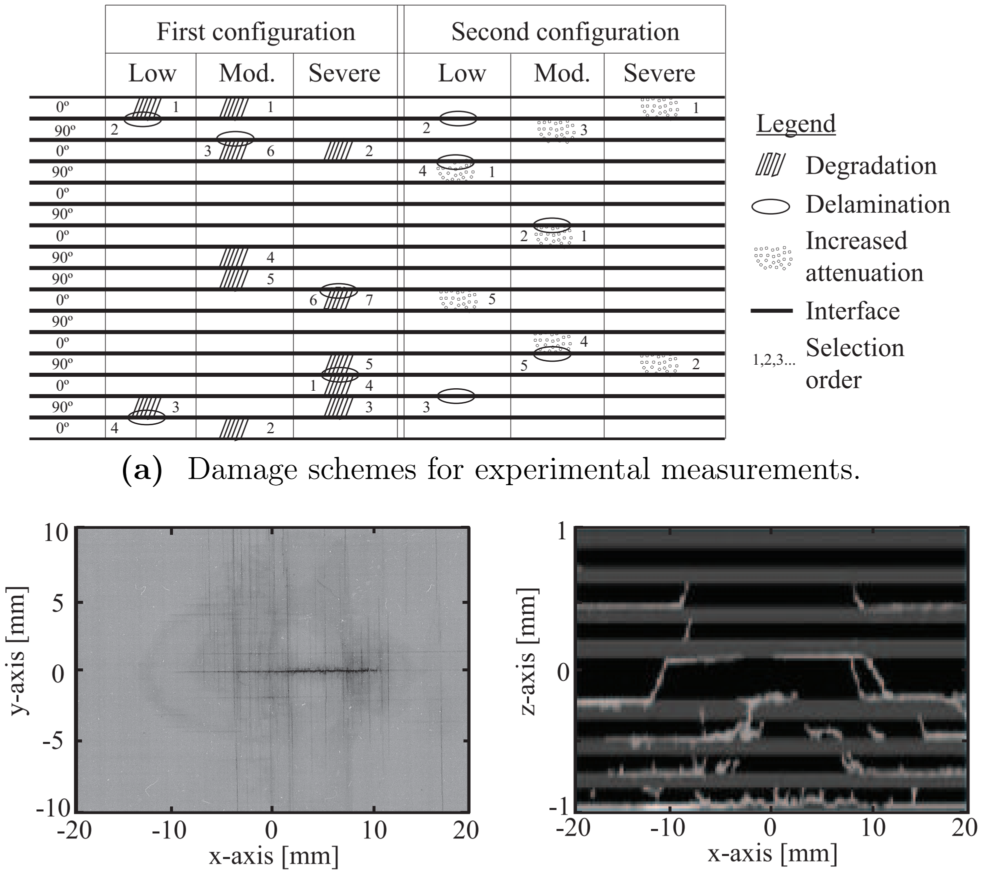
4.1 Discussion

Firstly, our blind damage parametrization algorithm was evaluated with three different synthetically damaged signals, which are related to impact damage distribution, as presented in Section 2.3, and they were assessed in our algorithm in order to know if it was able to capture the damaged layers and interfaces with three different levels of noise. As can be observed from the results, the parametrization is given in a proper way up to a level of noise of SNR = 20 dB, for which it starts to fail, as depicted in Test 3. Moreover, the relationship between the error percentage in the parameter reconstruction and the level of noise (Figure 3.8) showed a non-linear dependency, which was expected since many inverse problems are ill-posed: Solutions may not exist, they could be unstable and non-converging, or there may exist multiple solutions.

In addition, for the Test 2, by comparing Figure 3.6 with the model-class plausibilities in Table 3.6, it is clear that the first two parameters, in order of priority found by the sequential parametrization algorithm, are clearly discernible from undamaged (green marks) to damaged (red marks), which corresponds to parametrization plausibilities above 20%. In contrast, the subsequent parameters are still discernible but their uncertainties are comparable to the change of parameters values between undamaged to damaged states. The associated parametrization plausibilities are of the order of 10% or less. This behavior is similar to the two others synthetic Tests, where the first parameters have the higher plausibilities. Thus, the modeled signal tends to mimic the experimental one quickly in the first stages and then the others other stages are responsible for the matching of the few remaining signal echoes.

In the second part, experimental measurements taken from the damaged CFRP plate described in Section 2.1.1 were introduced in our blind algorithm obtaining results for two different reconstructions: with Young's modulus as variable in layers and interfaces in first place, and attenuation coefficients in layers added to Young's modulus in the interfaces in second place. As could be expected, with the first reconstruction, the parametrization increases its complexity with the level of damage. Thus, at low damage state our algorithm is able to mimic the measurement with 4 parameter whereas for higher damages it needs more parameters. Nonetheless, this behavior is the opposite with the second reconstruction since it decreases the number of parameters while increasing the damage complexity.

Furthermore, the evolution of the damage types and distribution across layers for the first configuration (see Figure 4.1 (a)) is compatible with micrograph observations (Figure 4.1 (b)). For instance, the first damaged layers are always degradation (stiffness reduction) of the boundary layers or close to them, followed by a strong delamination in the adjacent interfaces. This holds especially for low and moderate damage levels. At moderate damage, the central layers also degrade. Finally, at severe damage, the reconstruction is more random, which is coherent with the randomness of the evolution of strong damages inside a laminate. In particular, the re-



(a) Damage schemes for experimental measurements.

(b) CFRP plate images.

Figure 4.1 (a): Damage schemes within the CFRP plate thickness scheme with the order in the parameter selection in each position. (b): Radiograph and schematic of a micrograph of a cross-section for an impacted specimen removed after 100000 cycles of loading.

constructed damage is dominated by a degradation of layers or group of layers (and delamination) adjacent to the boundary ones, followed by a degradation and delamination of central layers.

In addition, model-class plausibilities can be seen in Table 3.13. By comparing both reconstruction, it can be concluded that first configuration yields more plausibility, in their first parameters, than the second one. This could be a reason for choosing the first configuration, since the BIP is solved with one parameter more differentiable, considering that it holds more plausibility, instead of the second one.

Interestingly, the strongest levels of damage are not prioritized by the algorithm. As an example, in the severe damage, first reconstruction, the almost full delimitation in layer 18 (99% stiffness reduction, which is the strongest reduction) is prioritized as one of the lasts most plausible. The second configuration, which includes attenuation, yields damage distributions less coherent with micrograph observations, which is compatible with the lower class plausibility. Notwithstanding, both configurations produce signal reconstructions which are far from perfectly mimic the measurements, as can be observed in Section 3.3.

4.2 Conclusions

The blind damage parametrization algorithm has been developed and evaluated with synthetic signals and experimental signals obtained from a damage CFRP plate. The best-suited model-class is selected by our algorithm in order to obtain a robust parametrization, which could explain the measurements. Moreover, we extract some concluding remarks of our novel blind parametrization algorithm as follows:

- The algorithm was able to reconstruct the synthetic signals with a remarkable degree of fitting, even though erroneous parameters were selected for the signal reconstruction and the high level of noise.
- The matching of the modeled signals with the experimental measurements is approximate, especially with the first configuration, since the signals are similar but not perfectly mimicked.
- The parameters were selected in a proper way for the synthetically damaged signals without noise and with a SNR = 25 dB. Higher levels of noise can produce erroneous choices in the selected parameters.
- The first parameters chosen by our algorithm were the most plausible

(more than 20%), whereas the following ones had plausibilities under 10%, which make less discernible these damaged parameters.

- The first configuration used in the reconstruction of the experimental signals ($E - E$ layers-interfaces) showed that the higher the damage level, the higher the number of parameters. In addition, the resulting parametrizations were positively contrasted against micrograph observations for composites.

- The second configuration ($\alpha - E$ layers-interfaces) showed damage pattern less coherent with micrograph observations. Furthermore, this configuration holds in most cases less plausibility than the first one.

Finally, we can promote some future trends in order to enhance this blind algorithm:

- The algorithm has to be algorithmically paralleled in order to reduce the computational cost, which could reduce to a few days the computation of an unique BIP.

- A deeper study of the attenuation coefficient as variable in the layers has to be addressed in order to remove some drawbacks like the large standard deviations obtained in this work.

- The 3D properties of the whole measured area of the CFRP plate should be reconstructed in order to provide complete damage maps for each layer/interface.

- New configurations, with a new set of variables (i.e. Young's moduli and attenuation coefficients in layers), have to be assessed in order to find a good set that does not compromise the computational cost and explains data with a high degree of precision.

Appendix A

Blind parametrization algorithm MATLAB code

```
% Sergio Cantero Chinchilla
% Damage hypothesis selection for US scan
% Bayesian Inverse Problem
% Compute the MCMC Metropolis—Hasting method

% Initialize variables for the BIP and UWP model
clear all; format compact; addpath([pwd '/lib']);
addpath([pwd '/lib_ip'  ]);
addpath([pwd '/lib_fpx' ]);

% Initialize algorithmic variables
cont_pos_vector=0;
pos_vector=[];
pos_in_la=[];
pos_med=[];
stop_cond=[];
GENERAL_STOP=false;

```

```
% Global loop
while cont_pos_vector<10 && GENERAL_STOP==false
        for x=31:-1:1

parameters_in=[2];
parameters_la=[2];

% Initialize variables for the BIP and UWP model

ip_vars;                      ip_init;
fpx_vars;                     fpx_init;
interfaces=[]; layers=[];
coincident=false;
k1=0;k2=0;k3=0;k4=0;

% Give the right positions of the current uncertain parameters
if length(pos_vector)>0
    for i=1:length(pos_vector)

        if pos_vector(i)==x
            coincident=true
            break
        end

        if pos_in_la(i)==true

            if rem(x,2)==0
                k1=k1+1;
                %layers=[];
                interfaces(1)=[x/2];
                interfaces=horzcat(interfaces,(pos_vector(i)/2));
                parameters=[2];
               %if length(layers)≥1
               %   parameters=horzcat(parameters,4);
```

```
                %end

            else
                k2=k2+1;
                layers(1)=[round(x/2)];
                interfaces(k2)=[pos_vector(i)/2];
                parameters=[2];
                %if length(interfaces)≥1
                %   parameters=horzcat(2,parameters);
                %end
            end

        else

            if rem(x,2)==0
                k3=k3+1;
                layers(k3)=[round(pos_vector(i)/2)];
                interfaces(1)=[x/2];
                parameters=[2];
                %if length(layers)≥1
                %   parameters=horzcat(parameters,4);
                %end

            else
                k4=k4+1;
                layers(1)=[round(x/2)];
                layers=horzcat(layers,round(pos_vector(i)/2));
                % interfaces=interfaces;
                parameters=[2];
                %if length(interfaces)≥1
                %   parameters=horzcat(2,parameters);
                %end

            end

        end
    end
else
             if rem(x,2)==0
                layers=[];
                interfaces=[x/2];
                parameters=[2];
                %if length(layers)≥1
                %    parameters=horzcat(parameters,4);
                %end

            else
                layers=[round(x/2)];
                interfaces=[];
                parameters=[2];
                %if length(interfaces)≥1
                %    parameters=horzcat(2,parameters);
                %end

            end
end


            if coincident==true
                Expinfgain(1:21,x)=0;
                Expctdlglikhd(x)=0;
                Logev(1:21,x)=0;
                continue
            end



% Initialize Factors of the M—H algorithm
```

```
std_mod_error = 0.48*.08;              % Std. dev. of the error

% Choose the configuration to finish MCMC M-H in function of
% the number of parameters

% 1 Parameter
if cont_pos_vector==0
   % Contator for plotting and burn-limit calculation:
   contstop=5000;
   % Cont for the calculation of the burn-limit:
   cont_burn_limit=30000;
   % First approximation of the fctr_prop (acceptance ratio?):
   cont_ini_fctr_prop=15000;
   % Final approx for the fctr_prop (Must be lower than cont_burn_limit):
   cont_fctr_prop=16000;
 if rem(x,2)==0
    fctr_prop=0.02;
   cont_burn_limit=30000;
   cont_ini_fctr_prop=20000;
   cont_fctr_prop=22000;
 else
     fctr_prop=0.03;
 end

 % 2 Parameters
elseif cont_pos_vector==1
   contstop=5000;
   cont_burn_limit=25000;
   cont_ini_fctr_prop=15000;
   cont_fctr_prop=20000;
   if rem(x,2)==0
       fctr_prop=0.01;
   cont_burn_limit=25000;
   cont_ini_fctr_prop=20000;
   cont_fctr_prop=22000;
   else
    fctr_prop=0.008;
   end

% 3 Parameters
elseif cont_pos_vector==2
   contstop=5000;
   cont_burn_limit=35000;
   cont_ini_fctr_prop=20000;
   cont_fctr_prop=30000;
   if rem(x,2)==0
       contstop=5000;
       cont_burn_limit=80000;
   cont_ini_fctr_prop=25000;
   cont_fctr_prop=27000;

    fctr_prop=0.012;
   else
    fctr_prop=0.009;
   end

% 4 Parameters
elseif cont_pos_vector==3
   contstop=10000;
   cont_burn_limit=60000;
   cont_ini_fctr_prop=40000;
   cont_fctr_prop=50000;
   if rem(x,2)==0
    fctr_prop=0.01;
    contstop=15000;
    cont_burn_limit=100000;
   cont_ini_fctr_prop=40000;
   cont_fctr_prop=45000;
   else
    fctr_prop=0.008;
   end

% 5 Parameters
elseif cont_pos_vector==4
   contstop=10000;
   cont_burn_limit=60000;
   cont_ini_fctr_prop=30000;
   cont_fctr_prop=45000;
   if rem(x,2)==0
       contstop=10000;
   cont_burn_limit=100000;
   cont_ini_fctr_prop=40000;
   cont_fctr_prop=50000;
    fctr_prop=0.009;
   else

    fctr_prop=0.005;
   end

% 6 Parameters
elseif cont_pos_vector==5
   contstop=10000;
   cont_burn_limit=90000;
   cont_ini_fctr_prop=30000;
   cont_fctr_prop=80000;
   if rem(x,2)==0
       contstop=10000;
   cont_burn_limit=120000;
   cont_ini_fctr_prop=50000;
   cont_fctr_prop=55000;
    fctr_prop=0.08;
   else
    fctr_prop=0.07;
   end

% 7 Parameters
elseif cont_pos_vector==6
   contstop=10000;
   cont_burn_limit=90000;
   cont_ini_fctr_prop=40000;
   cont_fctr_prop=70000;
   if rem(x,2)==0
       contstop=10000;
   cont_burn_limit=150000;
   cont_ini_fctr_prop=30000;
   cont_fctr_prop=31000;
    fctr_prop=0.06;
   else
    fctr_prop=0.06;
   end

% 8 Parameters
elseif cont_pos_vector==7
   contstop=20000;
   cont_burn_limit=120000;
   cont_ini_fctr_prop=60000;
   cont_fctr_prop=80000;
   if rem(x,2)==0
        contstop=50000;
   cont_burn_limit=360000;
   cont_ini_fctr_prop=340000;
   cont_fctr_prop=350000;
    fctr_prop=0.067;
   else
    fctr_prop=0.07;
   end

% 9 Parameters
elseif cont_pos_vector==8
   contstop=20000;
   cont_burn_limit=120000;
   cont_ini_fctr_prop=60000;
   cont_fctr_prop=80000;
   if rem(x,2)==0
   contstop=50000;
   cont_burn_limit=380000;
   cont_ini_fctr_prop=360000;
   cont_fctr_prop=370000;
    fctr_prop=0.055;
   else
    fctr_prop=0.06;
   end

% 9–10 Parameters
elseif cont_pos_vector>8
   contstop=20000;
   cont_burn_limit=130000;
   cont_ini_fctr_prop=100000;
   cont_fctr_prop=150000;
   if rem(x,2)==0
   contstop=50000;
   cont_burn_limit=380000;
   cont_ini_fctr_prop=360000;
   cont_fctr_prop=370000;
    fctr_prop=0.021;
   else
    fctr_prop=0.02;
   end
end
```

```

% Max number of MCMC steps (samples)
K=200000000;

n=[0:K/contstop];

% Delay between experiment signal and model signal
delay=8;

% Standard deviation of the prior PDF of the parameters
thck_std_prior=.01; E_std_prior=.4; Dnst_std_prior=.15;
Dmpng_std_prior=.6; Poiss_std_prior=.05;


%%%%%%%%%%%%%%%%%%%%%% METROPOLIS—HASTING ZONE %%%%%%%%%%%%%%%%%%%%%%

% Nominal values of the CFRP properties
[th thi Em Eeff rhom rhoeff dam num nueff model_m]=nominal_values(1);
model_m(:,4)=model_m(:,4)*1e—7;

% Extract the different nominal values (layers and interfaces)
Ela_pos=[];
Em_pos=[];

% Loop into the layers
pos=1;
pos2=1;
pos3=1;
for j=1:length(parameters_la)
    pos=pos;
    for i=1:length(layers)
        if cont_pos_vector>0 && (2*layers(i)—1)≠x
            th(pos,1)=pos_med(pos3);
            pos3=pos3+1;
        else
        th(pos,1)=model_m(2*layers(i)-1,parameters_la(j));
        end
        if parameters_la(j)==2
            Ela_pos(pos2)=pos;
            pos2=pos2+1;
        end
        pos=pos+1;
    end
end

% Loop into the interfaces
pos2=1;
for j=1:length(parameters_in)
    pos=pos;
    for i=1:length(interfaces)
        if cont_pos_vector>0 && (2*interfaces(i))≠x
            th(pos,1)=pos_med(pos3);
            pos3=pos3+1;
        else
            th(pos,1)=model_m(2*interfaces(i),parameters_in(j));
        end
        if parameters_in(j)==2
            Em_pos(pos2)=pos;
            pos2=pos2+1;
        end
        pos=pos+1;
    end
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%% LOG-NORMAL PDF PARAMETERS %%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Mu and sigma of Log—Normal PDF for each parameter
for i=1:length(parameters)
for j=1:length(th(:,1));

    if parameters(i)==1     % Thickness
    mu(j)=log(th(j,1)^2*1/sqrt(th(j,1)*thck_std_prior+th(j,1)^2*1));
    sigma(j)=sqrt(log(1+th(j,1)*thck_std_prior/th(j,1)^2*1));
    end

    if parameters(i)==2     % Young modulus E
    mu(j)=log(th(j,1)^2*1/sqrt(th(j,1)*E_std_prior+th(j,1)^2*1));
    sigma(j)=sqrt(log(1+th(j,1)*E_std_prior/th(j,1)^2*1));
    end

    if parameters(i)==3     % Density
    mu(j)=log(th(j,1)^2*1/sqrt(th(j,1)*Dnst_std_prior+th(j,1)^2*1));
    sigma(j)=sqrt(log(1+th(j,1)*Dnst_std_prior/th(j,1)^2*1));
    end

    if parameters(i)==4     % Damping
    mu(j)=log(th(j,1)^2*1/sqrt(th(j,1)*Dmpng_std_prior+th(j,1)^2*1));
    sigma(j)=sqrt(log(1+th(j,1)*Dmpng_std_prior/th(j,1)^2*1));
    end

    if parameters(i)==5     % Poison ratio
    mu(j)=log(th(j,1)^2*1/sqrt(th(j,1)*Poiss_std_prior+th(j,1)^2*1));
    sigma(j)=sqrt(log(1+th(j,1)*Poiss_std_prior/th(j,1)^2*1));
    end

end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% BAYES' THEOREM %%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Sampling the prior space for each parameter
for i=1:length(th)
prior_samples(:,i)=lognrnd(mu(i),sigma(i),[20000,1]);% Sampling generation
prctl_5(i)=prctile(prior_samples(:,i),5);            % Percentile 5%
prctl_95(i)=prctile(prior_samples(:,i),95);          % Percentile 95%
end

rng_param=abs(prctl_95-prctl_5);    % Space range

% Likelihood value for nominal values
lik=ip_fun(parameters_la,parameters_in,layers,interfaces,th(:,1),...
    model_m,std_mod_error,delay);

% Prior PDF
prior=prod(lognpdf(th(:,1)',mu,sigma));

% Posterior (Bayes' theorem)
target(1)=lik*prior;



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%% M-H ALGORITHM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Initialize values for the M-H algorithm
Exp_value=[eps];
acc_sum=0;
cont_n=1;
```

```
cont=0;
cont2=2;
cont3=0;
k=1;
stabilization=false;
cont_limit=0;
good_rate_ini=false;
good_rate=false;

% Start the loop
while stabilization==false ||good_rate_ini==false || good_rate==false

    % To know the step each 1000 steps
    k=k+1;
    if rem(k,1000)==0
    fprintf(['k=  ',num2str(k),'  parametro  ',num2str(x),...
             '  batida  ',num2str(cont_pos_vector+1),'\r\n'])
    end

    % Standard deviation of the proposal
    prop_stdv=fctr_prop⋆rng_param;

    % Proposal values (Normal distribution)
    th_prop=[mvnrnd(th(:,k−1)',prop_stdv.^2)];

    % Filtering wrong parameters

        for j=1:length(Ela_pos)
            if th_prop(Ela_pos(j))≥Eeff
                sep=th_prop(Ela_pos(j))−Eeff;
                th_prop(Ela_pos(j))=Eeff−sep;
            end
        end

        for j=1:length(Em_pos)
            if th_prop(Em_pos(j))≥Em
                sep=th_prop(Em_pos(j))−Em;
                 th_prop(Em_pos(j))=Em−sep;
            end
        end

        for j=1:length(th_prop)
            if th_prop(j)≤0
                th_prop(j)=0.001;
            end
        end
    % Likelihood value for proposal values

lik=ip_fun(parameters_la,parameters_in,layers,...
    interfaces,th_prop',model_m,std_mod_error,delay);



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%% METROPOLIS−HASTING TEST %%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    % Prior PDF (proposal values)
    prior=prod(lognpdf(th_prop,mu,sigma));

    % Posterior PDF (Bayes' Theorem)
    trgt_prop=lik*prior;

    % M−H ratio
    ratio=trgt_prop/target(k−1);

    % Value to compare with the random number
    alfa=min(1,ratio);
```

```

        % Random number comparison
        if alfa≥rand

            th(:,k)=th_prop';       % To store the proposal values
            target(k)=trgt_prop;    % To store the proposal PDF values
            cont=cont+1;            % Acceptance counter

        else

            th(:,k)=th(:,k−1);      % Take the previous parameters
            target(k)=target(k−1);  % Take the previous PDF proposal
        end

    % Cumulative acceptance rate
    acc_rate(k)=cont/k;         % Must be in [0.2,0.4];


    % To draw the acceptance ratio and the cumulative proposal mean
    if cont2==contstop

        k1=n(cont_n)*contstop+1;
        cont_n=cont_n+1;
        % Burn−in  and MH analysis
        %matlabpool open 6
        for i=k1:numel(target(1:k))  %parfor
            Exp_value(i)=(acc_sum+sum(target(k1:i)))/i;
        end
        %matlabpool close
        acc_sum=sum(target(:));


        cont2=0;    % Start again the counter to draw
        close all
```

```
                     %[left bottom width height]
    set(gcf,'PaperUnits','centimeters','PaperPosition',[1 1 18 9]);

    subplot(1,2,1)
    plot(Exp_value,'color','k')
    axis 'square'
    gca;

    subplot(1,2,2)
    plot(acc_rate,'color','k')
    axis 'square'
    print('-depsc')

   % Calculate the burn-period and the limit
   if k>cont_burn_limit
   cont_limit=cont_limit+1;
   [burn_period limit(cont_limit)]=burn_limit(Exp_value,...
                    cont_pos_vector,x,k,stabilization);

   % Estimate the convergence of the target PDF
   if cont_limit>2
        mean1=mean(Exp_value(limit(cont_limit-2):limit(cont_limit-1)));
        mean2=mean(Exp_value(limit(cont_limit-1):limit(cont_limit)));
        porc_stab=(mean2-mean1)/mean1;
         if abs(porc_stab)<.005
            stabilization=true;
            [burn_period limit(cont_limit)]=burn_limit(Exp_value,...
                         cont_pos_vector,x,k,stabilization);
         end
    end
   end
  end

% Accepntance rate checkout
```

```
cont2=cont2+1;


if rem(x,2)==0  % In interfaces

    % First checkout
    if k==cont_ini_fctr_prop && (acc_rate(k)>0.41 || acc_rate(k)<0.19)
        if acc_rate(k) > 0.81
            fctr_prop=fctr_prop*1.5;
            k=1;
            th(:,2:end)=[]; target(2:end)=[];
            cont_n=1; cont2=2; acc_sum=0; cont=0;
            clear Exp_value acc_rate
        elseif acc_rate(k) < 0.81 && acc_rate(k) > 0.5
            fctr_prop=fctr_prop*1.4;
            k=1;
            th(:,2:end)=[]; target(2:end)=[];
            cont_n=1; cont2=2; acc_sum=0; cont=0;
            clear Exp_value acc_rate
        elseif acc_rate(k) < 0.5 && acc_rate(k) > 0.4
            fctr_prop=fctr_prop*1.2;
            k=1;
            th(:,2:end)=[]; target(2:end)=[];
            cont_n=1; cont2=2; acc_sum=0; cont=0;
            clear Exp_value acc_rate
        elseif acc_rate(k) < 0.23 && acc_rate(k) > 0.1
            fctr_prop=fctr_prop*0.8;
            k=1;
            th(:,2:end)=[]; target(2:end)=[];
            cont_n=1; cont2=2; acc_sum=0; cont=0;
            clear Exp_value acc_rate
        elseif acc_rate(k) < 0.1
            fctr_prop=fctr_prop*0.6;
            k=1;
            th(:,2:end)=[]; target(2:end)=[];
            cont_n=1; cont2=2; acc_sum=0; cont=0;
            clear Exp_value acc_rate
        end

    else
       good_rate_ini=true;
    end

     % Last checkout
     if k==cont_fctr_prop && (acc_rate(k)>0.41 || acc_rate(k)<0.19)
        if acc_rate(k) > 0.81
            fctr_prop=fctr_prop*1.8;
            k=1;
            th(:,2:end)=[]; target(2:end)=[];
            cont_n=1; cont2=2; acc_sum=0; cont=0;
            clear Exp_value acc_rate
        elseif acc_rate(k) < 0.81 && acc_rate(k) > 0.6
            fctr_prop=fctr_prop*1.6;
            k=1;
            th(:,2:end)=[]; target(2:end)=[];
            cont_n=1; cont2=2; acc_sum=0; cont=0;
            clear Exp_value acc_rate
        elseif acc_rate(k) < 0.6 && acc_rate(k) > 0.39
            fctr_prop=fctr_prop*1.4;
            k=1;
            th(:,2:end)=[]; target(2:end)=[];
            cont_n=1; cont2=2; acc_sum=0; cont=0;
            clear Exp_value acc_rate
        elseif acc_rate(k) < 0.23 && acc_rate(k) > 0.1
            fctr_prop=fctr_prop*0.4;
            k=1;
            th(:,2:end)=[]; target(2:end)=[];
            cont_n=1; cont2=2; acc_sum=0; cont=0;
```

```
                clear Exp_value acc_rate
            elseif acc_rate(k) < 0.1
                fctr_prop=fctr_prop*0.2;
                k=1;
                th(:,2:end)=[]; target(2:end)=[];
                cont_n=1; cont2=2; acc_sum=0; cont=0;
                clear Exp_value acc_rate
            end
            else
           good_rate=true;
         end
else    % In layers
    % First checkout
    if k==cont_ini_fctr_prop && (acc_rate(k)>0.45 || acc_rate(k)<0.19)
        if acc_rate(k) > 0.8
            fctr_prop=fctr_prop*1.5;
            k=1;
            th(:,2:end)=[]; target(2:end)=[];
            cont_n=1; cont2=2; acc_sum=0; cont=0;
            clear Exp_value acc_rate
        elseif acc_rate(k) < 0.8 && acc_rate(k) > 0.6
            fctr_prop=fctr_prop*1.3;
            k=1;
            th(:,2:end)=[]; target(2:end)=[];
            cont_n=1; cont2=2; acc_sum=0; cont=0;
            clear Exp_value acc_rate
        elseif acc_rate(k) < 0.6 && acc_rate(k) > 0.4
            fctr_prop=fctr_prop*1.15;
            k=1;
            th(:,2:end)=[]; target(2:end)=[];
            cont_n=1; cont2=2; acc_sum=0; cont=0;
            clear Exp_value acc_rate
        elseif acc_rate(k) < 0.23 && acc_rate(k) > 0.1
            fctr_prop=fctr_prop*0.7;
            k=1;
            th(:,2:end)=[]; target(2:end)=[];
            cont_n=1; cont2=2; acc_sum=0; cont=0;
            clear Exp_value acc_rate
        elseif acc_rate(k) < 0.1
            fctr_prop=fctr_prop*0.5;
            k=1;
            th(:,2:end)=[]; target(2:end)=[];
            cont_n=1; cont2=2; acc_sum=0; cont=0;
            clear Exp_value acc_rate
        end

    else
       good_rate_ini=true; % Can go on
    end
     % Last checkout
     if k==cont_fctr_prop && (acc_rate(k)>0.39 || acc_rate(k)<0.05)
        if acc_rate(k) > 0.8
            fctr_prop=fctr_prop*1.8;
            k=1;
            th(:,2:end)=[]; target(2:end)=[];
            cont_n=1; cont2=2; acc_sum=0; cont=0;
            clear Exp_value acc_rate
        elseif acc_rate(k) < 0.8 && acc_rate(k) > 0.6
            fctr_prop=fctr_prop*1.6;
            k=1;
            th(:,2:end)=[]; target(2:end)=[];
            cont_n=1; cont2=2; acc_sum=0; cont=0;
            clear Exp_value acc_rate
        elseif acc_rate(k) < 0.6 && acc_rate(k) > 0.39
            fctr_prop=fctr_prop*1.4;
            k=1;
            th(:,2:end)=[]; target(2:end)=[];
            cont_n=1; cont2=2; acc_sum=0; cont=0;
```

```
            clear Exp_value acc_rate
        elseif acc_rate(k) < 0.23 && acc_rate(k) > 0.1
            fctr_prop=fctr_prop*0.4;
            k=1;
            th(:,2:end)=[]; target(2:end)=[];
            cont_n=1; cont2=2; acc_sum=0; cont=0;
            clear Exp_value acc_rate
        elseif acc_rate(k) < 0.1
            fctr_prop=fctr_prop*0.2;
            k=1;
            th(:,2:end)=[]; target(2:end)=[];
            cont_n=1; cont2=2; acc_sum=0; cont=0;
            clear Exp_value acc_rate
        end

    else
       good_rate=true; % Can go on
     end
end

end  % End the Metropolis-Hasting algorithm

limite=limit(end);
fprintf(['Acabo la iteracion:  ',num2str(x)],'\n')

% Evidence computation for model-class assessment
[Expinfgain(:,x) Expctdlglikhd(x) Logev(:,x)]=damage_evidence(delay,...
    interfaces,layers,model_m,mu,parameters_la,parameters_in,prop_stdv,...
    sigma,std_mod_error,prior_samples,target,th,burn_period,limit)

% Store the samples of the selected parameter
th_final{x}=th(:,burn_period:limite);

% Clear useless variables
```

```
clearvars -except  GENERAL_STOP stop_cond pos_med th_var th_final...
    Expctdlglikhd_var Prob_var Logev_var Expinfgain Expctdlglikhd Logev...
    z x pos_ch  pos_vector pos_in_la cont_pos_vector Expinfgain_var

end  % End the For that went through each layer and interface



% Position of the best parameter
    for i=1:length(Logev(end,:))
        if mean(Logev(:,i),1)==0
            Logev(:,i)=-1000000
        end
    end

    % Plausibilities of each Model-class
    med_logev=mean(Logev);
    for i=1:length(Logev)
        prob_post(i)=(exp(med_logev(i))*(1/length(Logev)))/...
            (sum(exp(med_logev))*(1/length(Logev)));
    end

    %pos_max=find(mean(Logev,1)==max(mean(Logev,1)));
    pos_max=find(prob_post==max(prob_post));

    aux_var=prob_post;
    aux_var(:,pos_max)=[];

    pos_max1=find(aux_var==max(aux_var));
    pos_ch=pos_max;

      % Layer of interface
      if rem(pos_ch,2)==0
          interfaz=true;
      else
          interfaz=false;
      end

      % Stop condition
      cont_pos_vector=cont_pos_vector+1;

      stop_cond(cont_pos_vector)=med_logev(pos_ch);
      if cont_pos_vector>1
          if stop_cond(cont_pos_vector)−stop_cond(cont_pos_vector−1)≤0
          GENERAL_STOP=true;
          end
      end


      % Store the position of the parameters
      pos_vector(cont_pos_vector)=pos_ch;

      % Store the information about layer or interface
      pos_in_la(cont_pos_vector)=interfaz;

      % Store the Expected Information Gained
      Expinfgain_var{cont_pos_vector}=Expinfgain;

      % Store the Expected Log−Likelihood
      Expctdlglikhd_var{cont_pos_vector}=Expctdlglikhd;

      % Store the Log−Evidence
      Logev_var{cont_pos_vector}=Logev;

      % Store the Plausibilities
      Prob_var{cont_pos_vector}=prob_post;

      % Store the samples of the selected parameter
      th_var{1}=th_final{pos_ch};

      % Store the mean of the samples of the selected parameters because it
      % will be the new mean of these parameters in the next stage
      pos_med=[];
      pos_med=mean(th_var{1,1},2);

      % Save the inforamtion for each stage
      save(['matrix_dimensions_424E_test_',num2str(cont_pos_vector)...
          ,'_batida.mat'],'pos_vector','Expinfgain_var',...
          'Expctdlglikhd_var','Logev_var','Prob_var','th_var',...
          'stop_cond','pos_med')

      % Clear the useless variables
      clear th_final pos_max aux_var pos_max1 pos_ch ...
          Expinfgain Expctdlglikhd Logev th_var
end
```

Appendix B

Subroutines

B.1 Likelihood function

```
% Likelihood function (cost functional )

function lik=ip_fun(parameters_la,parameters_in,layers,...
                    interfaces,th,model_m,std_mod_error,delay)
ip_vars;

% Residual (here you can define filters, calibrations)
residual =feval(nresidual,parameters_la,parameters_in,...
                layers,interfaces,th,model_m,delay);

    % Likelihood function
    for i=1:length(residual(end,:))
        % Cost functional (norm L2) (vectorized)
        f(i)   = -.5*sum(residual(:,i).^2);
        lik_signal(i) = exp(f(i)/std_mod_error^2);
    end
    lik=prod(lik_signal);

```

```

end
```

B.2 Residual

```
% Residual
function residual = fpx_residual(parameters_la,parameters_in,...
                                 layers,interfaces,th,model_m,delay)
fpx_vars;
ip_vars;

% Model (here can define parametrizations)
model  = feval(nmodel,parameters_la,parameters_in,layers,...
               interfaces,th,model_m);
msn   = feval(nmea,model,0);      % simulated observation
tmsn  = find(msn == max(msn),1);


for a=1:length(fpx_mod),
 fpx_modelxx=fpx_mod{a};
 eval(nexperiment);              % Load signal
 mxn=ip_meax(:);                 % Experimental observation
 tmxn = find(mxn == max(mxn),1); % Synchronize at maximum signal time
 msn_sync = interp1(1:length(msn),...
                    msn,tmsn-tmxn+(1:length(msn))','linear',0);
 residual(:,a) = mxn-msn_sync;    % Residual
end;

end
```

B.3 Definition of model parameters

```
%% Model definition

function model=model_def(parameters_la,parameters_in,...
                         layers,interfaces,th,model_m)
% Extract the different values

% Loop into the layers
pos=1;
for j=1:length(parameters_la)
    pos=pos;
    for i=1:length(layers)
        model_m(2*layers(i)-1,parameters_la(j))=th(pos);
        pos=pos+1;
    end
end
% Loop into the interfaces
for j=1:length(parameters_in)
    pos=pos;
    for i=1:length(interfaces)
        model_m(2*interfaces(i),parameters_in(j))=th(pos);
        pos=pos+1;
    end
end

%% Save structure data
model=struct('nam','fpx', 't',(0:.005e-6:9.995e-6)','m',model_m,
...
'freqc',5e6,'f','f5m','x','a05m1',
...
  'parn',{{'Thickness, {\it t} (mm)'
...
                  'Young modulus, {\it E} (GPa)'
...
```

```
                               'Density, \rho (kg/m^3)'
...
                                       'Rayleigh damping, \alpha'
...
                                               'Poisson ratio, \nu'}});

 end
```

B.4 Burn-in period calculation code

```

function [burn_period limit]=burn_limit(Exp_value,cont_pos_vector,...
                                        x,k,stabilization)


if stabilization==true
        %% With the mean
        if cont_pos_vector==0 || cont_pos_vector==1

            inc=10^cont_pos_vector;

        elseif cont_pos_vector==2
             if rem(x,2)==0
                inc=round(6^cont_pos_vector);
            else
                inc=round(3.5^cont_pos_vector);
             end
        elseif cont_pos_vector==3
            if rem(x,2)==0
                inc=round(6^cont_pos_vector);
            else
                inc=round(2^cont_pos_vector);
            end
            elseif cont_pos_vector>3
            if rem(x,2)==0
                inc=round(log2(k)*100);
            else
                inc=round(2^cont_pos_vector);
            end
        end

        burn_good=false;
        step=1;
        while burn_good==false
        %inc=round(log2(k)*100);
        cont=0;
        for i=1:inc:length(Exp_value)
            cont=cont+1;
            if (i+inc)<length(Exp_value)
                med_int(cont)=mean(Exp_value(inc:i+inc));
            else
                med_int(cont)=med_int(cont-1);
            end
        end

        for i=1:length(med_int)-1
            med_int1(i)=abs((med_int(i+1)-med_int(i))/med_int(i+1));
        end

        for i=length(med_int1):-1:1
        if med_int1(i)>0.001 && (i+1)*inc<length(Exp_value)
        burn_period=(i+1)*inc
        break
        end
        end

        %% With the slope
```

```

        inc1=20;
        cont=0;
        for i=1:inc1:length(Exp_value)
            %i
           cont=cont+1;
            if  (i+inc1)<length(Exp_value)
                grad(cont)=(Exp_value(i+inc1)−Exp_value(i))/(inc1);
            else
                grad(cont)=grad(cont−1);
            end
        end

        % thre_burn=max(grad)/(.5*abs(log(max(Exp_value1))));
        thre_lim=min(grad)/1000;

        cont=0;
        for i=1:length(grad)−1

            if grad(i+1)−grad(i)<thre_lim
                limit=(i+1)*inc1;
            end

        end



        % If failed here once more, try to increment
        % this inc in a thrid hypothetic step in this while.

        if step<3 || burn_period<(limit)
            if burn_period<(limit)
                burn_good=true;
            else
                if inc≤3
```

```
                    inc=1; burn_good=true;
                else
                inc=round(inc/3);
                end
                clear grad med_int med_int1
            end
        elseif burn_period≥(limit)
             if inc≤3
                    inc=1; burn_good=true;
                else
                inc=round(inc*200);
                end
                clear grad med_int med_int1
        end
        step=step+1;
        end


else
    %% With the slope

        inc1=20;
        cont=0;
        for i=1:inc1:length(Exp_value)
           cont=cont+1;
            if (i+inc1)<length(Exp_value)
                grad(cont)=(Exp_value(i+inc1)-Exp_value(i))/(inc1);
            else
                grad(cont)=grad(cont-1);
            end
        end

        % thre_burn=max(grad)/(.5*abs(log(max(Exp_value1))));
        thre_lim=min(grad)/1000;

        cont=0;
        for i=1:length(grad)-1

            if grad(i+1)-grad(i)<thre_lim
                limit=(i+1)*inc1;
            end

        end
        burn_period=0;
end

end
```

B.5 Model-class evidence computation code

```
% Code to compute the evidence of a model class definition


function [Expinfgain Expctdlglikhd Logev]=damage_evidence(delay,...
    interfaces,layers,model_m,mu,parameters_la,parameters_in,...
    prop_stdv,sigma,std_mod_error,prior_samples,target,th,burn,limit)

ip_vars;                      ip_init;
fpx_vars;                     fpx_init;
contsave=1;
cont=1;
%% Definition of the parameters to compute the model class evidence
% Definition of the amount of parameters saved

% Must be between burn and limit (must be close to the limit)
% This will be the "center value" of the PDF
if 0.6*limit(end)>burn
```

```
    cyc_cent1=[0.6*limit(end):(.35*limit(end)/20):.95*limit(end)];
elseif 0.6*limit(end)<burn && 0.79*limit(end)>burn
    cyc_cent1=[0.8*limit(end):(.15*limit(end)/20):.95*limit(end)];
else
    cyc_cent1=[(burn+1):((1*limit(end)-(burn+1))/20):1*limit(end)];
end

cyc_cent=round(cyc_cent1);
N2=1000;                         % Number of samples
N1=limit(end)-burn+1;            % Number of samples

for j=1:length(cyc_cent)
j
% Selected parameters. Parameter "k"
th_slct=th(:,burn:limit(end));
% Target (proposal) values of the selected parameters
trgt_slct=target(burn:limit(end));

for  i=1:N1
    %i
    if j==1
    % Prior values of the selected parameters
    prior_slct(i)=prod(lognpdf(th_slct(:,i),mu',sigma'));
    % Likelihood values of the selected parameters as target/prior
    lik_slct(i)=trgt_slct(i)./prior_slct(i);
    end

    if i==1
        % Selected parameters around the central value
        th_cent(:,j)=th_slct(:,cyc_cent(j)-burn);
        % Target (proposal) values of the central values
        trgt_cent(j)=target(cyc_cent(j)-burn);
        % Prior values of the central parameters
        prior_th(j)=prod(lognpdf(th_cent(:,j),mu',sigma'));
        % Likelihood values of the central parameters as target/prior
        lik_cent(j)=trgt_cent(j)./prior_th(j);
    end

    %% Calculation of average goodness of fit
    if i==N1
        %Log−Likelihood of the selected parameters
        Expctdlglikhd=(1/N1)*sum(log10(lik_slct(:)));
    end
    %% Calculation of the analytical approx of posterior

    % Calculation of numerator of analytical approx. of posterior
    % Evaluation of the proposal, centred in the selected parameters
    % and evaluating the central values
    eps_k(i)=mvnpdf(th_cent(:,j)',th_slct(:,i)',diag(prop_stdv.^2));

    % Evaluation of the ratio between central parameters target
    % and selected parameters target
    tau_k(i)=min(1,trgt_cent(j)/trgt_slct(i));

end

% Numerator of the approx posterior
numerator(j)=1/N1*sum(eps_k.*tau_k);

% Calculation of denominator
% Proposed samples, N2 samples
prop_samples=[mvnrnd(th_cent(:,j),prop_stdv.^2,N2)];

for l=1:length(prop_samples(end,:))
for i=1:length(prop_samples(:,end))
       if prop_samples(i,l)≤0
          prop_samples(i,l)=0.001;
       end
```

```
end
end

for i=1:N2
    % Prior pdf of the samples
    prior(i)=prod(lognpdf(prop_samples(i,:),mu,sigma));
    lik(i)=ip_fun(parameters_la,parameters_in,layers,...
           interfaces,prop_samples(i,:),...
           model_m,std_mod_error,delay); % Likelihood of the samples

    % Target (proposal) of the proposed samples
    target_l(i)=prior(i)*lik(i);

    % Ratio between target of proposed values
    % and the target of central value
    rho_l(i)=min(1,target_l(i)/trgt_cent(j));

end
% Denominator of de approx posterior
denominator(j)=1/N2*sum(rho_l);
% Posterior approximate
post_approx(j)=numerator(j)/denominator(j);

%% Log-evidence for a model class
Logev(j)=log10(lik_cent(j))+log10(prior_th(j))-log10(post_approx(j));

%% Expected information gained
Expinfgain(j)=Expctdlglikhd-Logev(j);

cont=cont+1

Expinfgain_mean=mean(Expinfgain)

Expctdlglikhd
```

```

Logev_mean=mean(Logev)
end

end
```

List of Figures

List of Tables

Bibliography

[1] C. Bakis, L. C. Bank, V. Brown, E. Cosenza, J. Davalos, J. Lesko, A. Machida, S. Rizkalla, T. Triantafillou, Fiber-reinforced polymer composites for construction-state-of-the-art review, *Journal of Composites for Construction* 6 (2) (2002) 73–87.

[2] L. Hollaway, A review of the present and future utilisation of FRP composites in the civil infrastructure with reference to their important in-service properties, *Construction and Building Materials* 24 (12) (2010) 2419–2445.

[3] S. Abrate, Impact on laminated composites: recent advances, *Applied Mechanics Reviews* 47 (11) (1994) 517–544.

[4] R. Talreja, Damage and fatigue in composites–a personal account, *Composites Science and Technology* 68 (13) (2008) 2585–2591.

[5] W. Cantwell, P. Curtis, J. Morton, Impact and subsequent fatigue damage growth in carbon fibre laminates, *International journal of fatigue* 6 (2) (1984) 113–118.

[6] W. Cantwell, P. Curtis, J. Morton, Post-impact fatigue performance of carbon fibre laminates with non-woven and mixed-woven layers, *Composites* 14 (3) (1983) 301–305.

[7] N. Uda, K. Ono, K. Kunoo, Compression fatigue failure of CFRP lami-

nates with impact damage, *Composites Science and Technology* 69 (14) (2009) 2308–2314.

[8] Q. Shen, M. Omar, S. Dongri, Ultrasonic NDE Techniques for Impact Damage Inspection on CFRP Laminates., *Journal of materials science research* 1 (1).

[9] F. Aymerich, S. Meili, Ultrasonic evaluation of matrix damage in impacted composite laminates, *Composites Part B: Engineering* 31 (1) (2000) 1–6.

[10] A. Tarantola, *Inverse problem theory and methods for model parameter estimation*, siam, 2005.

[11] V. K. Kinra, C. Zhu, Ultrasonic nondestructive evaluation of thin (subwavelength) coatings, *The Journal of the Acoustical Society of America* 93 (5) (1993) 2454–2467.

[12] V. Kinra, P. Jaminet, C. Zhu, V. Iyer, Simultaneous measurement of the acoustical properties of a thin-layered medium: The inverse problem, *The Journal of the Acoustical Society of America* 95 (6) (1994) 3059–3074.

[13] V. K. Kinra, V. R. Iyer, Ultrasonic measurement of the thickness, phase velocity, density or attenuation of a thin-viscoelastic plate. Part II: the inverse problem, *Ultrasonics* 33 (2) (1995) 111–122.

[14] M. J. Maron, R. J. Lopez, *Numerical analysis: a practical approach*, Macmillan New York, 1982.

[15] K. Balasubramaniam, S. C. Whitney, Ultrasonic through-transmission characterization of thick fibre-reinforced composites, *Ndt & E International* 29 (4) (1996) 225–236.

[16] K. Balasubramaniam, N. S. Rao, Inversion of composite material elastic constants from ultrasonic bulk wave phase velocity data using genetic algorithms, *Composites Part B: Engineering* 29 (2) (1998) 171–180.

[17] A. A. Fahim, R. Gallego, N. Bochud, G. Rus, Model-based damage reconstruction in composites from ultrasound transmission, *Composites Part B: Engineering* 45 (1) (2013) 50–62.

[18] J. L. Beck, Bayesian system identification based on probability logic, *Structural Control and Health Monitoring* 17 (2010) 825–847.

[19] M. Chiachio, J. Chiachio, A. Saxena, G. Rus, K. Goebel, Fatigue damage prognosis in frp composites by combining multi-scale degradation fault modes in an uncertainty Bayesian framework, *Proceedings of the 9th international workshop on structural health monitoring, stanford, ca, USA, 10-12 september* .

[20] J. Chiachio, M. Chiachio, A. Saxena, G. Rus, K. Goebel, Bayesian model selection and parameter estimation for fatigue damage progression models in composites, *International Journal of Fatigue* .

[21] A. Fahim, R. Gallego, N. Bochud, G. Rus, Model-based damage reconstruction in composites from ultrasound transmission, *Composites Part B: Engineering* 45(1) (2013) 50–62.

[22] S.-C. Wooh, I. M. Daniel, Three-dimensional ultrasonic imaging of defects and damage in composite materials, *Materials Evaluation;*(United States) 52 (10).

[23] M. Hosur, C. Murthy, T. Ramamurthy, A. Shet, Estimation of impact-induced damage in CFRR laminates through ultrasonic imaging, *NDT & E International* 31 (5) (1998) 359–374.

[24] N. Bochud, *Signal processing-based identification of pathology using ultrasonics*, Ph.D. thesis, University of Granada, 2014.

[25] H. Schmutzler, M. Alder, N. Kosmann, H. Wittich, K. Schulte, Degradation monitoring of impact damaged carbon fibre reinforced polymers under fatigue loading with pulse phase thermography, *Composites Part B: Engineering* 59 (2014) 221–229.

[26] T. Mitrevski, I. Marshall, R. Thomson, The influence of impactor shape on the damage to composite laminates, *Composite Structures* 76 (1) (2006) 116–122.

[27] R. Demirli, J. Saniie, Model-based estimation of ultrasonic echoes. Part I: Analysis and algorithms, *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on* 48 (3) (2001) 787–802.

[28] S. H. Cheung, J. L. Beck, Calculation of posterior probabilities for Bayesian model class assessment and averaging from posterior samples based on dynamic system data, *Computer-Aided Civil and Infrastructure Engineering* 25 (5) (2010) 304–321.

[29] E. Jaynes, *Papers on probability, statistics and statistical physics*, (Ed. R.D Rosenkrantz),Kluwer Academic Publishers, 1983.

[30] E. Jaynes, *Probability Theory: The Logic of Science*, Cambridge University Press, 2003.

[31] R. Neal, Probabilistic inference using Markov chain Monte Carlo methods, *Intelligence* 62.

[32] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, Equation of state calculations by fast computing machines, *The* J*ournal of* C*hemical* P*hysics* 21 (1953) 1087–1092.

[33] W. K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, *Biometrika* 57 (1) (1970) 97–109.

[34] D. An, J.-H. Choi, N. H. Kim, S. Pattabhiraman, Fatigue life prediction based on Bayesian approach to incorporate field data into probability model, *Structural engineering & mechanics* 37 (4) (2011) 427.

[35] D. Lunn, D. Spiegelhalter, A. Thomas, N. Best, The BUGS project: Evolution, critique and future directions, *Statistics in medicine* 28 (25) (2009) 3049–3067.

[36] E. T. Jaynes, Information theory and statistical mechanics, *Physical review* 106 (4) (1957) 620.

[37] J. L. Beck, K. V. Yuen, Model selection using response measurements: Bayesian probabilistic approach, *Journal of Engineering Mechanics* 130 (2004) 192.

[38] M. Muto, J. L. Beck, Bayesian updating and model class selection for hysteretic structural models using stochastic simulation, *Journal of Vibration and Control* 14 (1-2) (2008) 7–34.

[39] S. Kullback, R. A. Leibler, On Information and Sufficiency, *The* A*nnals of* M*athematical* S*tatistics* 22 (1) (1951) 79–86.

[40] H. W. Engl, M. Hanke, A. Neubauer, *Regularization of inverse problems*, vol. 375, Springer, 1996.