

A numerically-enhanced machine learning approach to damage diagnosis using a Lamb wave sensing network

C. Sbarufatti ^{a,*}, G. Manson ^b, K. Worden ^b

^a Politecnico di Milano, Dipartimento di Meccanica, Via La Masa 1, 20156 Milano, Italy

^b Dynamics Research Group, Department of Mechanical Engineering, University of Sheffield, Mappin Street, Sheffield S1 3JD, UK

Article history:

Received 26 February 2013

Received in revised form

22 March 2014

Accepted 25 April 2014

Handling Editor: I. Trendafilova

Available online 27 May 2014

1. Introduction

Structural Health Monitoring (SHM) as a technology offers the promise of reduced cost-of-ownership and increased safety of operation for a wide range of engineering structures. The main advantage of implementing SHM is that it allows a move towards condition-based maintenance. More fundamentally, SHM will facilitate a move from safe-life design principles to damage tolerant ones; the potential result being a reduction in conservative reserve factors, allowing the design and build of lighter greener structures. However, the widespread practical implementation of SHM is still some way away. One of the reasons for this situation is that SHM cannot currently deliver the level of diagnostic information desired by industry with confidence. In the specific case of data-based SHM, one of the main problems is the lack of availability of

* Corresponding author. Tel.: +39 02 2399 8213.

E-mail address: claudio.sbarufatti@polimi.it (C. Sbarufatti).

data from damaged structures. Pure data-based SHM makes use of pattern recognition or machine learning methods in the attempt to diagnose structural condition from measured data without recourse to physics-based structural models [1].

Machine learning algorithms fall broadly into two classes: *unsupervised learning* and *supervised learning* approaches. Supervised learning approaches develop a classifier based on the availability of training data that can assign a class label to unseen data; in the context of SHM, this class label could indicate the location or severity of damage or could simply indicate whether the structure is damaged or not. The issue with supervised learning is that the training data must include examples from all classes of interest. In the SHM context, the training data would need to include data from structures with all conceivable damage locations and severities for example. For very high-value structures like aircraft or bridges, such data could not be collected by experiment; the cost would be inconceivable. This leaves the possibility that the training data be collected from a programme of numerical simulation; this also has problems in that models of high enough fidelity for complex structures are impossible to build or could be very expensive. In the absence of damage data, one can pursue *unsupervised learning*. In this approach one uses only data from the undamaged structure in order to build a statistical model of normal condition data; one subsequently tests new data in order to see if it is consistent with the model and any significant deviation is taken as an indication of damage. This approach has problems also (see [1]; Chapter 12), but the main issue is that it really only allows detection of damage, it does not usually deliver higher level diagnostic information like location or severity of damage. If one needs higher level diagnostics, one is forced to adopt supervised learning or abandon the machine learning approach.

As collection of training data for supervised learning by experimental methods is generally impossible, one must address the issue by building physical law-based models, or where possible by referring to surrogate damage such as the application of lumped masses acting as a perturbation of the nominal baseline condition [2–4]. Supervised learning approaches have been used for many years for diagnostic purposes and the possibility to learn from models has a long history too [5,6]. Nevertheless, there has been very little success reported in the literature in developing diagnostic systems trained on simulated data that could generalise well to the real situation. A model-based diagnostic system based on strain field monitoring was presented in [7], providing satisfactory results for anomaly detection, localisation and quantification when used in real time to diagnose a real crack propagating on an aluminium stiffened skin structure. Further noteworthy work is reported in [8,9], where the authors tested a model-based diagnostic system based on Lamb wave scattering for the quantitative diagnosis of through-hole-type defects in a composite quasi-isotropic laminate.

In a model-based SHM framework it is important to stress the attention on the numerical verification and validation with experimental signals in order to provide reliable network outputs. Additionally, as numerical information is generally free from noise and environmental influences (sometimes included if training patterns are experimentally generated) it is also important to verify the system sensitivity to input variations through propagation of artificial uncertainties. Finally, during SHM system design, one has to guarantee sufficient generalisation performance, which is a non-trivial aspect to be considered when pure numerical data are provided as training examples. The modest aim of the current paper is to extend knowledge on how a combination of model-based and data-based SHM can be accomplished in a realistic, but rather idealised, experimental context.

The structure considered in this paper is a simple plate and the objective of the SHM system designed is to locate and quantify the extent of a crack in the plate. A machine learning approach is adopted based on artificial neural networks. The basic data used for diagnostic purposes are Lamb wave time histories for waves transmitted across the plate and thus scattered by any damage. Simulated Lamb wave signals have been used to extract scalar damage indices thus generating a database of numerical experience, describing the sensitivity of the selected feature to crack damage in different positions and with various extents. After verification and validation of numerical experience, damage indices from different paths across the plate are assembled into a feature vector which is used as the basis for two neural network regressors (one for localisation, the other for quantification). The optimisation of the algorithm structure is a crucial aspect, especially when a model which is trained on simulated experience has to generalise on real data. The strategy for network training adopted here is that many networks are trained for both the location and severity assessment and they are then assembled into a *committee* structure. A statistical procedure for the selection of the best performing network structure has been used, based on *Analysis of Variance*. Then, a method inspired by *Information Gap Theory* and based on *interval arithmetic* has been used to propagate uncertainties through the network, with the aim of appreciating the sensitivity to input variations. The system performance has finally been validated with real experimental Lamb wave signals. The main objective of the paper is to show a realistic case study where the data-based approach is enhanced by the availability of a physics-based model that can be used to generate training data. The main outcome of this study is the definition of a methodology for the robust design, verification and validation of a hybrid data/model-based SHM system.

The layout of the paper is as follows: a brief introduction to Lamb wave signal modelling and a summary of the proposed diagnostic methodology steps are reported in Sections 2 and 3 respectively. The experimental setup is described in Section 4. Lamb wave modelling and validation are explained in detail in Section 5, while the diagnostic algorithm structure definition and training are described in Section 6, where a method for the evaluation of network robustness in the simulated environment is also adopted. Finally, a description of the test results when real experimental data are processed into the diagnostic algorithms is presented in Section 7, which is followed by a conclusions section.

2. Elastic Lamb wave propagation modelling

Some basic principles necessary to understand the theory behind the analytical and numerical models for Lamb wave propagation are reported in this section. It is not the purpose of the present paper to explain in detail the mathematics

behind Lamb wave propagation; the interested reader could refer to [10–12] for a detailed analysis of the elastic wave propagation. A proper description of the numerical model adopted to simulate Lamb waves can also be found in the literature [13–15].

2.1. The analytical solution

If wave propagation for an *unbounded medium* is concerned (no boundary conditions influence the problem), two different propagating waves can be established, namely *longitudinal* (or *pressure*) and *transverse* (or *shear*) waves and the following considerations arise: (i) velocities only depend on material constants because the medium is considered infinite, thus no influence of geometry appears; (ii) the two types of wave do not interact with each other; (iii) wave velocities are independent from wave frequency, thus they are called *non-dispersive* and (iv) the adoption of Lamè constants implies an isotropic material. As a consequence of the latter, the wave velocity is independent of the wave propagation direction.

If a *bounded medium* is considered, the situation is much more complicated. Consider, as an example, the case of a solid medium bounded by two parallel planes at a distance $d = 2h$ apart (Fig. 1a). This case resembles the specimen adopted in the experimental part of this work. The solution of the elastodynamic problem of wave propagation provides two types of solution, one referred to as symmetric, the other anti-symmetric (Fig. 1b).

In particular, after imposing null stress conditions at the boundaries, the following non-trivial solutions are obtained for the symmetric and anti-symmetric modes respectively:

$$\frac{\tan(qh)}{\tan(ph)} = -\frac{4k^2pq}{(q^2 - k^2)^2} \quad (1)$$

$$\frac{\tan(ph)}{\tan(qh)} = -\frac{4k^2pq}{(q^2 - k^2)^2} \quad (2)$$

where

$$p = \frac{\omega}{\sqrt{V_l^2 - c^2}}, \quad q = \frac{\omega}{\sqrt{V_t^2 - c^2}} \quad (3)$$

k is the wavenumber, which specifies the number of oscillations per unit of space and V_l and V_t are the longitudinal and transverse wave velocities for the unbounded medium respectively. In particular, c is the target wave velocity that needs to be estimated. It is clear that given a value of frequency ω , Eqs. (1) and (2) specify the allowed value for c . This implies that waves propagating in bounded media are *dispersive*. In particular, the velocity depends on the product ωh . Moreover, Eqs. (1) and (2) may have any number of real solutions depending on the assumed value for ωh . The waveforms associated to different velocities are termed *modes* and can be divided into symmetric and anti-symmetric modes. It is often convenient to show the velocity of the modes as a function of the frequency-thickness product. Such graphs are named *dispersion curves* and are very helpful for preliminary optimisation of guided-wave SHM systems. Dispersion curves are reported in Fig. 2 for a 2 mm thick aluminium skin with Young's modulus $E = 74,000$ MPa, density $\rho = 26$ kg/dm³ and Poisson's ratio $\nu = 0.33$. In order to completely define a propagating wave, dispersion curves are available for both phase velocity and group velocity. The former is the rate at which the phase of the wave propagates in space while the latter is the velocity of the signal envelope in space. It is clear that, the higher the selected operating frequencies, the more propagating modes will be involved. Those modes will interact with each other thus making much more complicated the signal simulation and interpretation. As will be described later, it is common practice to generate a test wave packet characterised by relatively low frequencies, in order to *cut-off* the higher modes, thus taking into consideration only the two fundamental modes during signal actuation (the S_0 , fundamental symmetric mode and the A_0 , fundamental anti-symmetric mode). Nevertheless, it is

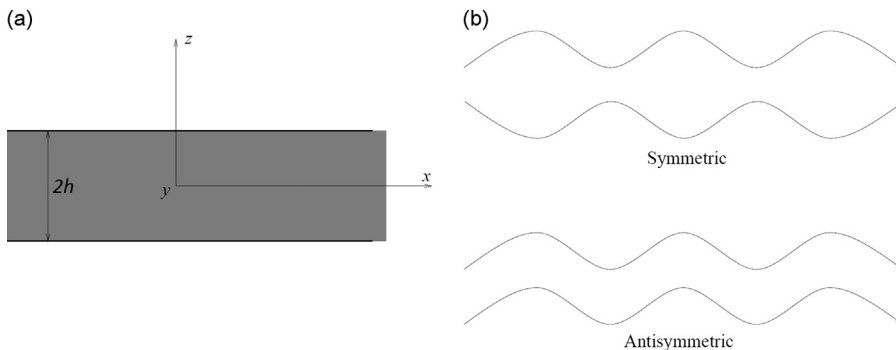


Fig. 1. (a) Coordinate system for Lamb waves in a plate and examples of (b) symmetric and (c) anti-symmetric Lamb wave modes.

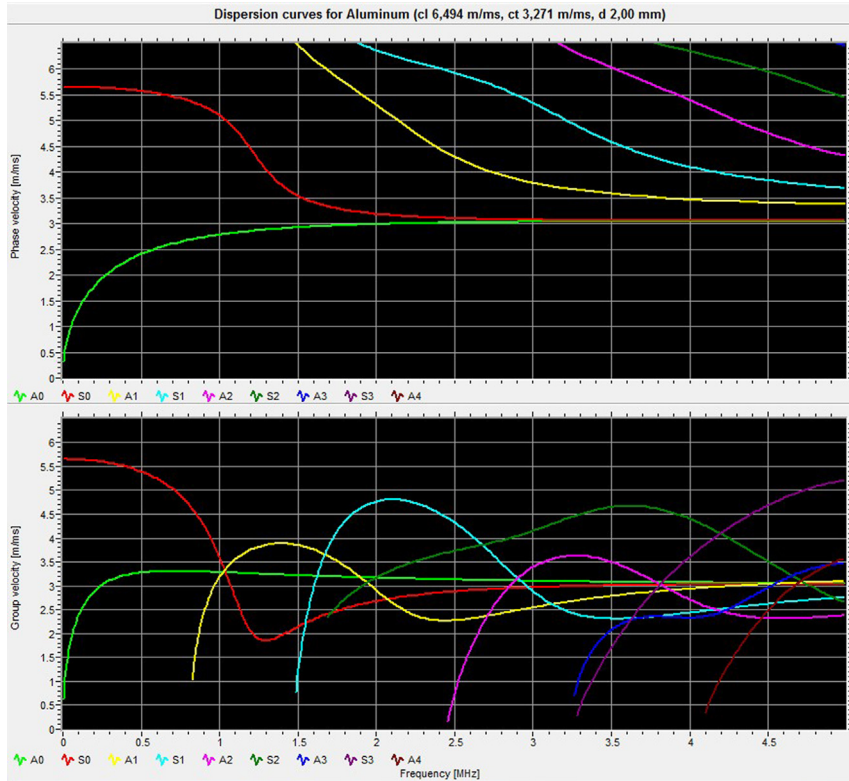


Fig. 2. Dispersion curves for an aluminium plate with 2 mm thickness, as calculated with *Valen Dispersion* [16] software. Symmetric and anti-symmetric mode velocities have been reported in terms of phase and group velocities as a function of signal frequency.

important to consider that, even though one single mode is initially propagated (either A_0 or S_0), the wave interaction with either the damage or the geometry boundaries may cause mode conversion as well as reflection and transmission.

The analytical model, represented by dispersion curve plots, can be used to estimate the propagation velocity of a wave packet, thus evaluating the a priori expected arrival time at sensors for the same wave packet. As will be shown in detail in [Section 5](#), this allows focusing on a particular time window of the acquired signals.

2.2. The numerical solution

The theoretical solution of the problem of ultrasonic guided wave propagation in complex structures is not easy and often impossible to obtain. Numerical modelling is one answer to this problem. Different numerical computation techniques can be used for the analysis of elastic wave propagation. In particular, Finite Difference (FD) equations and Finite Element (FE) analysis have long histories and have been used for wave propagation modelling for decades. Both consist of numerical methods which provide an approximate solution of the differential equations governing the elastic problem. FE theory is based upon a discretisation of the medium and attempts to find the exact solution in correspondence to the mesh nodes. The solution extrapolated where nodes do not exist is approximated through shape functions which can be arbitrarily chosen depending on the complexity of the problem to be analysed. FD methods approximate the function derivatives using local variables on a small grid spacing. The solution is only available in correspondence to the grid nodes and it is based on the direct approximation of the partial differential equations that describe the problem itself. One can allow higher order derivatives in order to provide a more accurate solution to the investigated problem.

FE analysis is particularly well suited to problems with very complex geometries. On the other hand FD-based methods are convenient for wave propagation in media with homogeneous or continuously varying physical properties. Nevertheless, the presence of boundaries or discontinuities between different types of media can lead to approximate FD solutions and can produce severe errors. The more recent Local Interaction Simulation Approach (LISA) was proposed to avoid these difficulties. The method is formally similar to the FD approach and uses the so-called Sharp Interface Model (SIM) imposing the continuity of displacement and stresses at interfaces and discontinuities. In contrast to classical FD-based techniques, which directly simulate a physical model of wave propagation, the LISA/SIM method does not put the mathematical equations associated with the problem into boundaries. In practice this method uses only three boundary conditions: free reflection, fixed reflection and absorption at the end. The absorption at the end depends on the model configuration used.

LISA can provide accurate description of wave propagation [17,18] and diffusion problems even in complex media. A complete description of the LISA method is provided in [13–15].

Concerning the work reported hereafter, a comparison of commercial ABAQUS FE code with LISA code is provided in Section 5.1. The LISA approach has then been selected and used throughout the work, being particularly apt to run on a Graphic Processing Unit (GPU) capable of high computational parallelisation (CULISA).

3. The methodology

A flowchart of the methodology adopted in this study is presented in Fig. 3. The overall procedure, explained in detail in the following sections of this article, can be divided into four main parts: Lamb wave signal modelling, creation of the numerical experience database, definition of the algorithm structure and the final experimental validation of the entire procedure.

The first requirement of the methodology is the availability of information about the baseline Lamb wave signal relative to the undamaged case. This information is found here by analysis and by computational modelling. In the first case, the analytically derived dispersion curves for Lamb wave propagation in aluminium [11,12] are used in order to determine the wave velocities and thus their arrival time at given sensors. In the second case, the Local Interaction Simulation Approach (LISA) [13–15] is used to simulate the entire waveform propagation. Once the agreement between the arrival times analytically calculated and experimentally evaluated has been verified, the analytical (dispersion curves) and the LISA models can be used to focus on a particular window in the time domain relative to selected wave packets, chosen according to their sensitivity to damage. In order to generate a database of damage cases as required by the supervised learning approach, the LISA model is given a parametric dependence on the position and size of a crack in the plate and is used to generate data corresponding to many crack locations and sizes. In the study presented here, the crack angle was kept constant, although the method still remains valid when this additional parameter needs to be considered.

When a machine learning approach is used, one must choose to examine a data feature which is both sensitive to the damage and of low dimension. This means extracting and condensing the damage sensitive information from the time data in the form of some sort of damage index. Choice of this index is critical to the success of the classifier. In the current paper, the damage indices are constructed by cross-correlating the Lamb wave signals with baseline signals simulated from the undamaged case; the correlation index thus gives a measure of ‘distance from normality’. One can compute a damage index in this way for any propagation path between an actuator and sensor and the set of indices so formed is used as a vector input to the damage classifier. The whole set of training data is constructed by simulating all the possible damage cases and computing the damage indices one would observe in each case relative to the simulated undamaged system.

Once the database of damage indices has been numerically derived as a function of crack length and position, it is given as training data to an Artificial Neural Network (ANN) structure, which should then be able to infer the location and dimension of damage when presented with new experimental data. In order to increase the robustness of the inference procedure, a network committee is trained which provides as output the average of many different ANN trained structures. The procedure for optimisation and training of the ANN structure is described in detail in Section 6 of this paper. The availability of a numerical database has another advantage, consisting in the possibility to predict the network performances in the simulated environment before processing the real experimental data into the diagnostic algorithm. A technique inspired by Information Gap Theory and based on interval arithmetic has been used here to propagate some uncertainty in the input through the network, allowing an appreciation of output sensitivity.

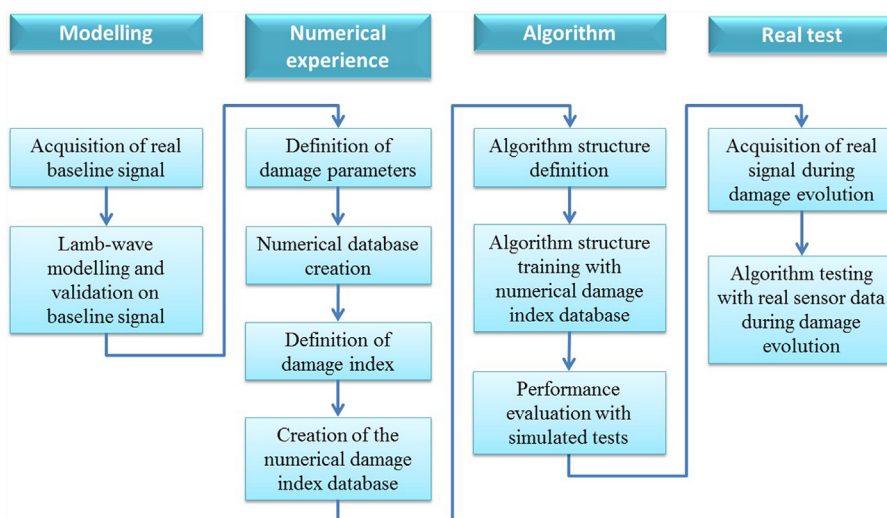


Fig. 3. Flowchart representing the procedure to realise and test the numerically enhanced signal processing.

Table 2
Path definitions (refer to Fig. 4 for actuator/sensor ID reference).

Path ID	Actuator ID	Sensor ID	Type	Path ID	Actuator ID	Sensor ID	Type	Path ID	Actuator ID	Sensor ID	Type
1	1	5	I	11	2	5	II	21	5	6	III
2	5	1	I	12	5	2	II	22	6	5	III
3	2	4	I	13	3	6	II	23	1	3	IV
4	4	2	I	14	6	3	II	24	3	1	IV
5	2	6	I	15	1	2	III	25	4	6	IV
6	6	2	I	16	2	1	III	26	6	4	IV
7	3	5	I	17	2	3	III	27	1	6	V
8	5	3	I	18	3	2	III	28	6	1	V
9	1	4	II	19	4	5	III	29	3	4	V
10	4	1	II	20	5	4	III	30	4	3	V

following Section 4.4. The damage index set is given as input to the damage classifier, which is trained on numerical simulated experience, thus providing an estimation of crack length and position.

4.4. Experimental procedure

Considering the sensor network is composed of six sensors, and that each sensor can act both as both actuator and sensor, a total number of 30 *pitch-catch* paths can be defined and used to scan the aluminium skin highlighting the presence of any anomaly. Hereafter, the calculation of a certain variable (e.g. a damage index) for the totality of the available paths is referred to as a *map* of that variable. Paths have been reported in Table 2, where they have been divided into five categories as follows:

- Short diagonal paths (Type I),
- Horizontal paths (Type II),
- Short vertical paths (Type III),
- Long vertical paths (Type IV), and
- Long diagonal paths (Type V).

Each path map was obtained by sequentially scanning all the available paths by connecting each piezo-actuator to the waveform generator and each relative sensor to the oscilloscope. In particular, each Lamb wave signal recorded is the result of an average operated on 15 consecutive excitations of the same actuator. A first acquisition of the Lamb wave signal map was performed on the undamaged skin, to be used as a reference for the validation of the analytical (dispersion curve) and numerical (LISA) models. Lamb waves were launched by driving the actuator with a 5-cycle toneburst, modulated with a Hanning window. Various frequencies were initially considered, as described in Section 5.2.1, in order to evaluate and validate the dispersion curve plots as well as to select the most apt frequency for the situation under monitoring. There then followed a second map acquisition for the healthy structure, to be used for diagnostic purposes. After the damage was artificially induced into the structure, a Lamb wave signal map was acquired at each stage of crack length reported in Table 1. The signals were then processed through the algorithm described in Section 6.

5. Lamb wave modelling and validation

5.1. The numerical model

The experimental panel shown in Fig. 4 was numerically modelled in the current study using three different software codes, two based on a Finite Difference concept (LISA and CULISA) and one based on Finite Element theory (ABAQUS). A brief comparison of the three methods is presented here, focusing the attention on the required computational time. As a matter of fact, the entire methodology requires a massive number of simulations to build the numerical experience necessary to train the signal processing algorithm for structural diagnosis, as will be described later.

The panel material was taken to be aluminium with the following properties: Young's modulus $E=74,000$ MPa, density $\rho=26$ kg/dm³, Poisson's ratio $\nu=0.33$, bulk longitudinal wave velocity $V_l=6.494$ mm/ μ s and bulk shear wave velocity $V_t=3.271$ mm/ μ s.

In order to avoid modelling the piezoelectric elements as well as the interaction between the sensors/actuators and the aluminium skin, a point force excitation was selected as the best compromise between accuracy of the solution and required computational time. The actuation for the A_0 mode excitation was modelled as two input forces orthogonal to the skin plane, one applied on top of the panel and one on the bottom (Fig. 5a). The actuator for the S_0 mode excitation was modelled as two distributed radial forces parallel to the skin plane and directed along the required paths, applied at the top and bottom of the skin (Fig. 5b). The displacements in the direction orthogonal to the plate were simulated/measured and stored at each sensor position.

Focussing on the LISA method; according to the literature review [19], a minimum number of 4 elements through the plate thickness (2 mm) is recommended to obtain a reliable solution, which constrains the nominal dimension of the mesh down to 0.5 mm. A total number of 1.4 million cubes were needed to model the structural part. The time increment for the simulation of Lamb wave propagation was set to $0.05 \mu\text{s}$ in order to ensure that the simulation be numerically stable [17].

Concerning the application of Finite Element analysis for Lamb wave simulation, the recommendations reported in [20] were considered. A maximum mesh dimension of 0.67 mm and a maximum time increment of $0.1 \mu\text{s}$ were specified in order to guarantee the stability of an ABAQUS EXPLICIT code. As these limits are compatible with those associated with LISA theory, the latter was used for both the models.

The effect due to the adoption of a different number of elements through the specimen thickness can be appreciated inside Fig. 6 as a function of the considered Lamb wave mode (either A_0 or S_0) and the chosen modelling software. Only LISA and ABAQUS have been considered concerning the evaluation of model accuracy as CULISA returns exactly the same output as LISA, it being the same software but optimised in terms of computational performance. It is clear that the choice of a sufficient number of computational cells through the thickness is critical, especially when the A_0 mode is considered. Moreover, a discrepancy was found between LISA and ABAQUS; while the adoption of a coarse mesh provokes a slightly early arrival time with the LISA code, it generates a strong delay with the ABAQUS FEM software. However, both softwares were found to tend to the analytical solution while improving the mesh size. Considering the S_0 mode, no significant influence of the mesh was recorded, at least relative to the time window corresponding to the arrival of the first wave packet. This is reflected in an almost perfect superposition of the signals calculated with the two codes during the excitation of the S_0 mode. A non-negligible phase shift is highlighted when comparing the A_0 modes evaluated with FEM and LISA, nevertheless they are expected to approach each other if a mesh refinement is allowed. As a result of this preliminary evaluation, the usage of simulated experience based on the S_0 mode appears to be more robust and reliable for the diagnosis of the monitored structure.

The results in terms of required computational time are reported in Table 3 for the three tested numerical methods. In particular, one should notice the extreme advantage due to the adoption of CUDA parallel computing on a Graphical Processing Unit (GPU), which allowed a reduction of more than 10 times the computational requirements for one simulation [21]. Given that the proposed methodology requires running many simulations (in the order of thousands) to build up the experience for the algorithm training, CULISA was adopted for the creation of the SHM training database, as described in Section 5.3.

5.2. Validation of undamaged numerical model

The validation of the LISA model described in Section 5.1 is the subject matter of the following sections. In particular, four validation levels are presented as follows:

- Level 1: dispersion curve validation,
- Level 2: signal map validation for the undamaged skin,
- Level 3: model validation in the presence of growing damage, and
- Level 4: damage index parameter validation.

Levels 1 and 2 are presented in the current section, dealing with the undamaged situation, while levels 3 and 4 are discussed in the Section 5.3, which is focused on the simulated damage experience database.

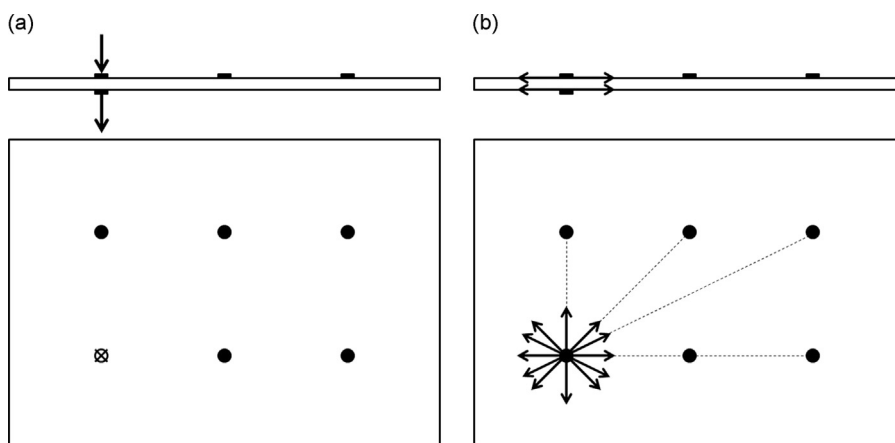


Fig. 5. Scheme of the excitation model for the A_0 and S_0 mode, respectively in (a) and (b).

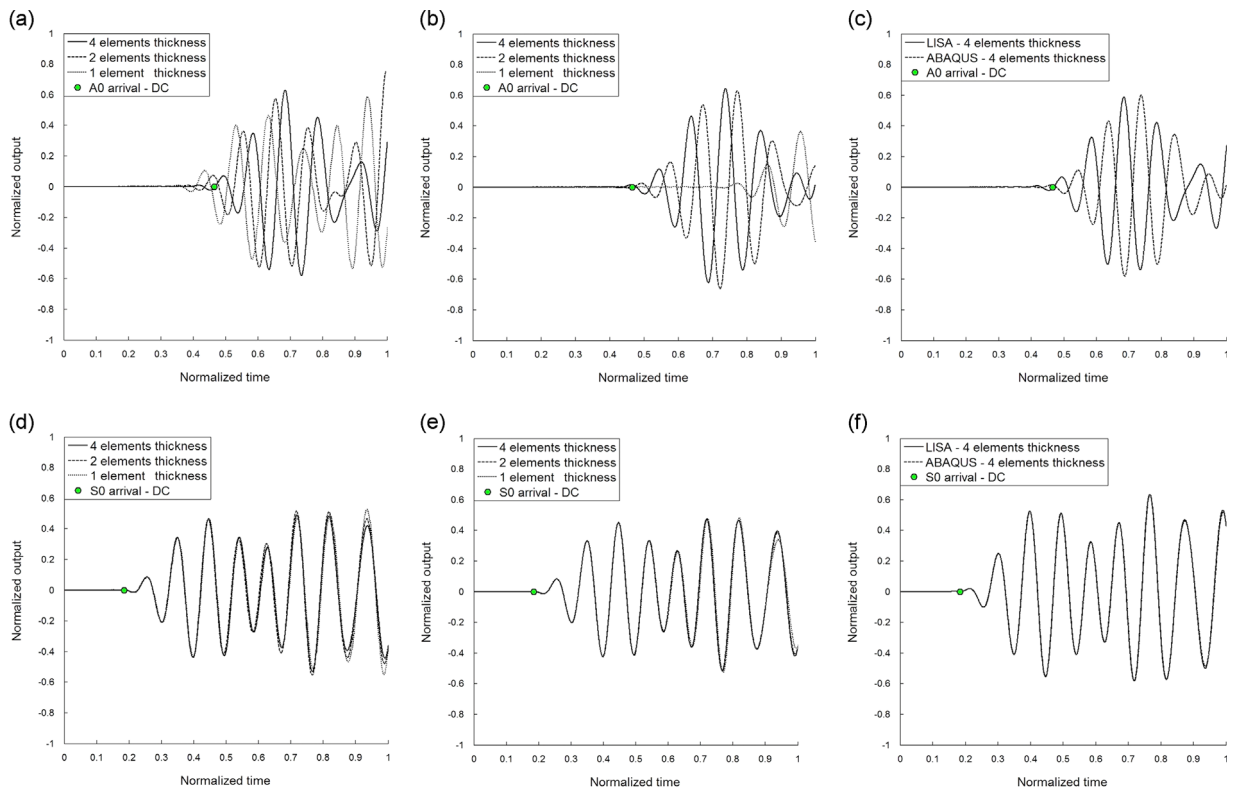


Fig. 6. Simulated Lamb wave signals for A_0 (up) and S_0 (down) mode calculated with (a,d) LISA (b,e) ABAQUS and (c,f) comparison between the two models. The estimation of the first wave packet arrival calculated with dispersion curves is also provided in each sub-figure. The same path is considered in each picture, where the displacements in the direction orthogonal to the plate have been measured.

Table 3

Comparison of computational time requirements. Results are normalised with respect to LISA time duration.

SW code	Duration of one simulation
LISA	1
CULISA	1/13
ABAQUS	3/4

5.2.1. Dispersion curves

A comparison of simulated and experimental data including information from the dispersion curves is presented in Fig. 7; the results are for the single path with the actuator in position 1 and the sensor in position 5 (as defined in Fig. 4). Two simulated signals are shown, representing the cases when only single A_0 or S_0 modes are excited, as described above. In contrast, the experimental signals contain both A_0 and S_0 modes. The analytical calculation of the wave packet arrival time, based on the dispersion curves, is also provided as a term of comparison in each sub-figure.

Four frequencies were examined inside the range below the first cutoff frequency (800 kHz as visible in Fig. 2) in order to avoid exciting higher Lamb wave modes (only A_0 and S_0 are present). The aim was both to appreciate the validity of the used model and to select the most apt frequency to be used for the structural monitoring system under examination. As a matter of fact, a decomposition of the signal in its A_0 and S_0 components is very useful in order to facilitate the creation of the Lamb wave propagation model and for selection of damage sensitive features. The idea is to consider the Lamb wave signal only contained within the first wave packet windows associated with the S_0 and A_0 mode arrivals. In particular, one could use the arrival time of a mode (analytically calculated) as the initial window limit and the duration of the tone burst (simulated) as the window time span. If possible, the superposition of the two modes in one window should be avoided in order to allow for the decomposition of the signal into its A_0 and S_0 parts.

Considering Fig. 7, it is clear that the arrival times of the A_0 and S_0 modes respectively decrease and increase with frequency; this means that the adoption of a lower frequency would allow for a larger time without mode superposition. On the other hand, one must consider that the wave packet associated with lower frequencies is characterised by a larger time span. This can also generate signal complexity as reflections of the first wave packet from the plate boundaries can arrive

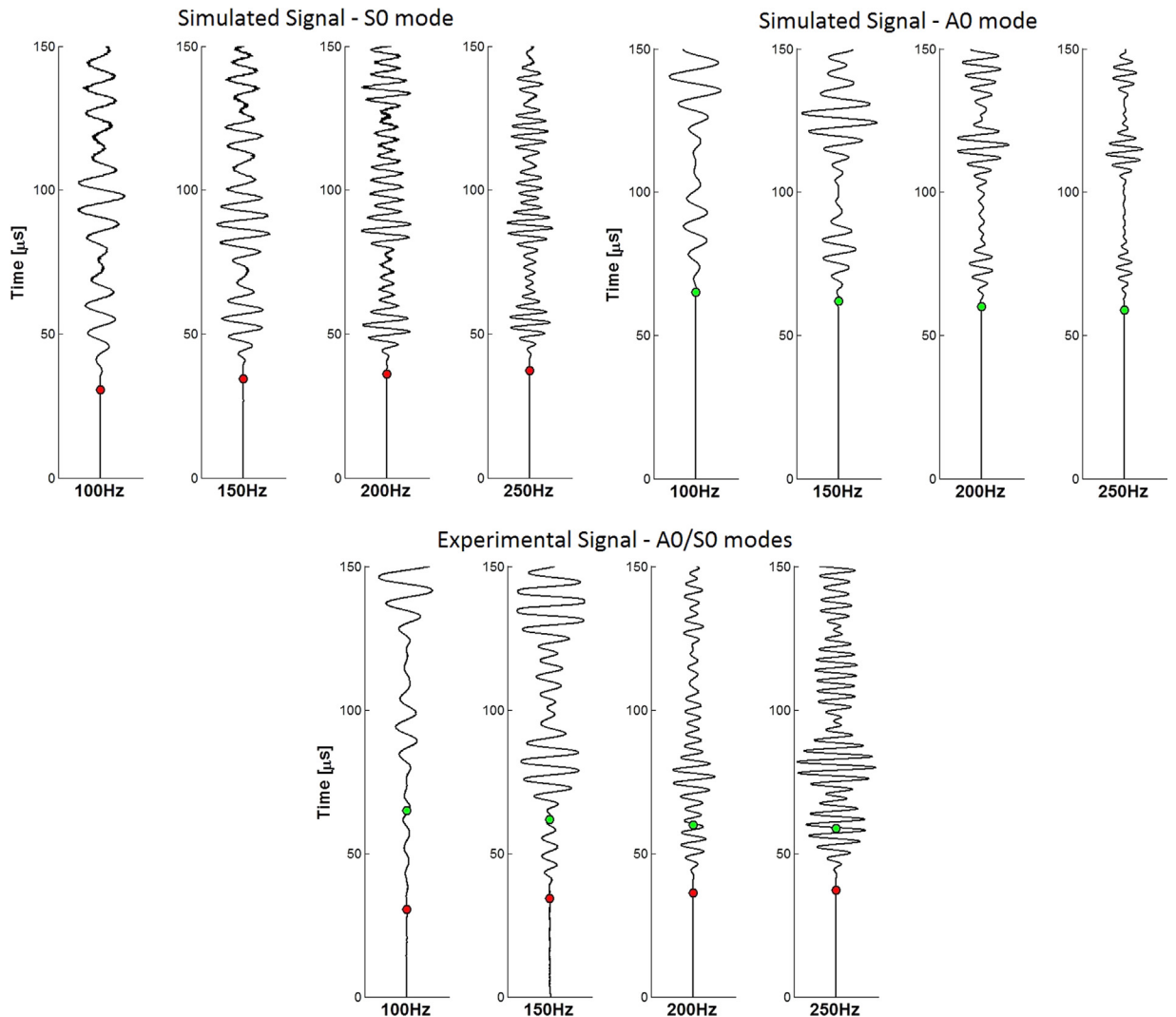


Fig. 7. Comparison of analytical, numerical and experimental data. The analytical solution for wave packet arrival time is represented through circles (red circle for S_0 arrival and green circle for A_0 arrival). The experimental signal comprehends both A_0 and S_0 propagation modes, while they have been represented separately for the numerical case. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

before the end of the considered acquisition window. Moreover, higher frequencies are critical from the point of view of numerical computational time as they require a smaller time increment step during simulation; 150 Hz was chosen as a compromise frequency and will be used for the methodology development hereafter.

Concerning modelling accuracy, the arrival times of the Lamb wave propagation modes in the simulations agree very well with those computed using the dispersion curves. A strong consistency can also be appreciated between the numerical and experimental signals at various frequency levels. The dispersion curves are confirmed to be a valuable reference for the selection of the most interesting time window from either the numerically simulated or experimentally acquired signals.

5.2.2. Undamaged model validation

The comparison between the numerically simulated and experimentally acquired signals is presented in Figs. 8 and 9 for the S_0 and A_0 modes respectively. The undamaged structure is considered here and all the available paths have been considered, divided into the five categories defined in Table 2. The signals relative to similar paths have been averaged for both the numerical and experimental cases, allowing for a clear comparison of the results. The arrival times of the S_0 and A_0 modes evaluated through the dispersion curves are also provided in order to identify the beginning of the first wave packets for the modes under consideration.

Considering the S_0 mode (Fig. 8), an almost perfect agreement in terms of signal phase was found between the numerical solution and the experimental data, particularly for the relatively short paths (types I, II and III). However, some delays were encountered in the numerical solutions regarding the longer paths (types IV and V). Only the time window relative to the first wave packet should be considered as the numerical solution only contains information about the S_0 mode while the experimental signal contains both the zero modes; this results in almost complete disagreement between the numerical and experimental data in the remainder of the signal. Some discrepancies have been found for the signal envelopes, nevertheless one could mitigate their influences through a correct choice of the damage index, as will be discussed later.

Focussing on the A_0 mode (Fig. 9) it is important to recall that the associated experimental first wave packet might contain a component of the S_0 mode (the S_0 mode arrival time is always shorter than the A_0 one), while the numerical simulation only contains information about the A_0 mode. Even so, a good agreement between the numerical and experimental data is found for path types I, II and III, in terms of both the phase and the envelope of the first A_0 wave packet. A poor numerical description was obtained for paths of types IV and V, due to the large influence of S_0 components (S_0 boundary reflections are included also) in the time window associated to the first A_0 wave packet.

The adoption of the S_0 mode is thus recommended for the creation of the simulated experience necessary to train the SHM algorithms for structural diagnosis. Again, it is important to notice that the analytical calculation of the arrival times is in complete agreement with both the numerical and experimental solutions and thus might be used for the selection of the signal portion which is most interesting for diagnostic purposes.

5.3. The simulated experience database

As the undamaged baseline model is now considered to be successfully validated, the next step of the proposed methodology consists in creating the simulated experience database to be used for algorithm training, as will be described inside Section 6. The diagnostic variables considered here are both crack length and position and the aim is to build a reasoning tool able to diagnose such damage inside the sensor grid presented in Fig. 4. The simulated crack centre positions to be considered are shown in Fig. 10, together with the sensor grid and the position of the real damage centre induced during the experiment. A series of damage lengths have been modelled at each selected damage position, starting from 5 mm, and increasing to 60 mm in 5 mm increment steps. The damage has been modelled here by assigning the material properties of air to the mesh cubes in correspondence with the notch. The entire map of signals needs to be acquired for each damage case, thus requiring a number of runs equal to the available actuator number. The same procedure has to be repeated for all the modes that need to be considered (although the SHM system performance will be presented only for the S_0 mode processing, a preliminary evaluation of both A_0 and S_0 modes was performed). A summary of the required simulations is reported in Table 4; this required a global simulation time of 10 days on two parallel machines DELL Precision T5500 2.13 GHz running CUDA QUADRO FX4800 with 1.5 GB memory.

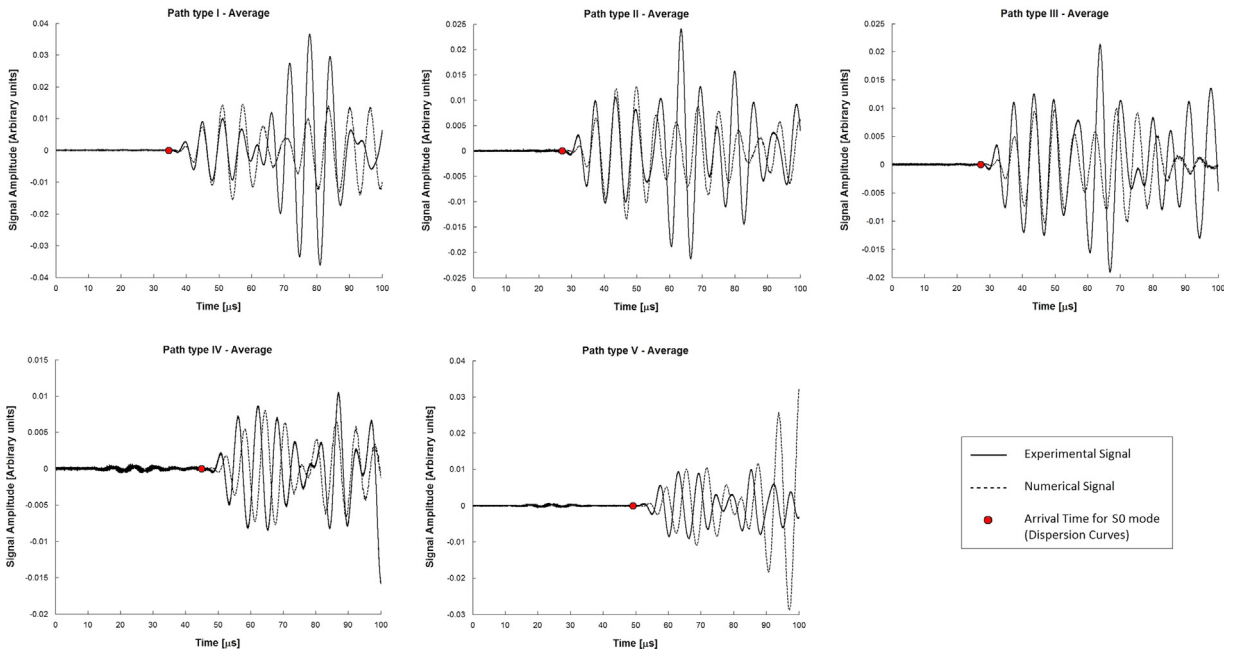


Fig. 8. Comparison of numerical (S_0 mode) and experimental signals relative to the entire path map. The arrival time of S_0 mode, analytically calculated with dispersion curves, is also marked. Refer to Table 2 for path type definition.

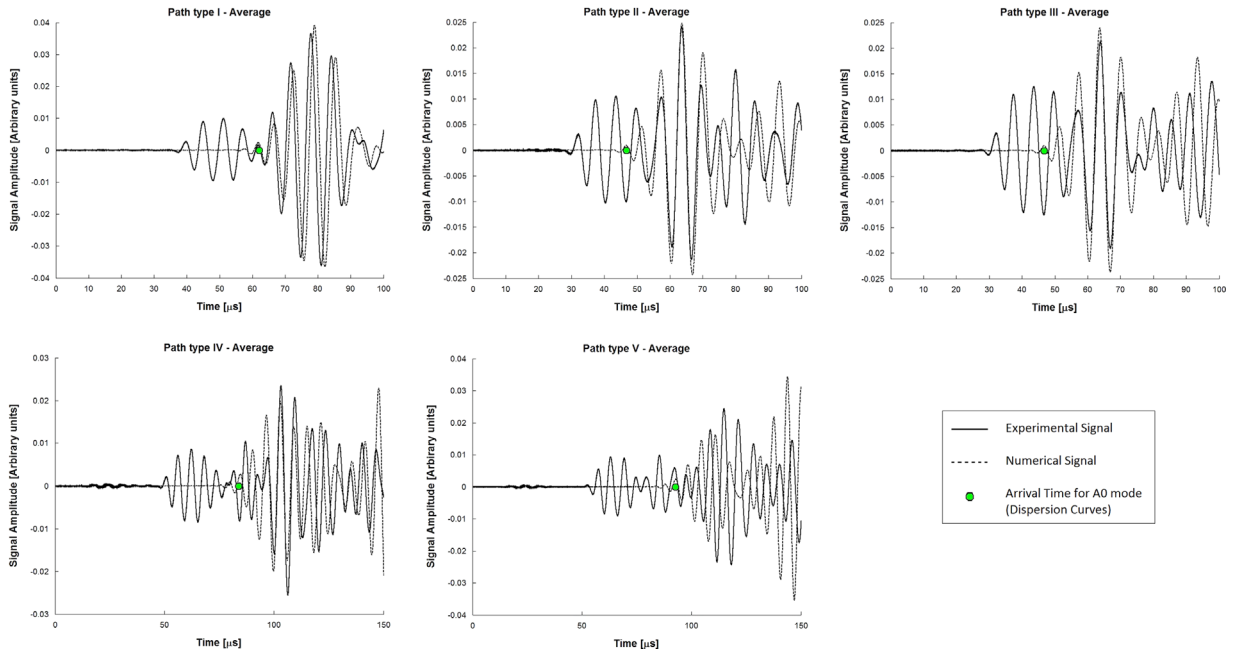


Fig. 9. Comparison of numerical (A_0 mode) and experimental signals relative to the entire path map. The arrival time of A_0 mode, analytically calculated with dispersion curves, is also marked. Refer to Table 2 for path type definition.

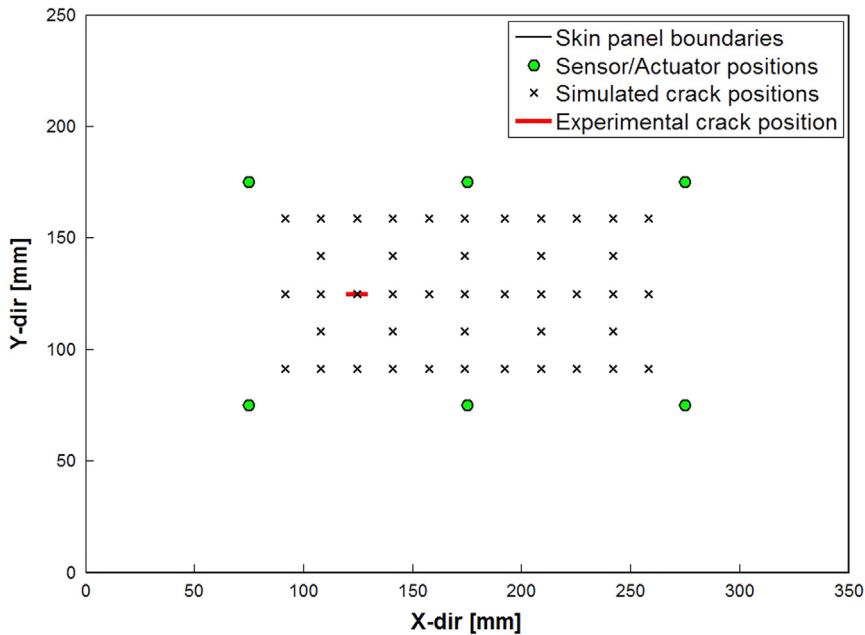


Fig. 10. Crack centre positions used for the creation of the simulated experience database.

5.3.1. Signal dependence on crack length

A comparison of the numerical and experimental signal sensitivities to crack length are given in Figs. 11 and 13 respectively for the S_0 and A_0 first wave packets. Only the path associated with actuator 1 and sensor 5 has been represented, and the numerically simulated damage corresponds to the real damage induced during the experiments (Fig. 4). The time span of the selected windows corresponds to about $33 \mu\text{s}$, as the frequency of the 5-cycle toneburst is 150 kHz.

Considering the S_0 mode (Fig. 11), an overall dependency of both the numerical (a) and experimental (b) signals on the crack length is encountered. The presence of the damage induces a monotonic reduction of the signal amplitude as well as a monotonically increasing delay in the wave packet arrival. This information can be exploited for structural diagnosis. Some considerations can be made to improve the performances of the SHM reasoning unit. Firstly, according to the analytical

dispersion curves, the arrival time of the A_0 mode happens inside the time window relative to the S_0 first wave packet. It is possible to cutoff the selected window in correspondence with this limit in order to avoid any A_0 disturbances. A further improvement can be obtained avoiding the signal complexity due to S_0 reflections. Referring to the particular case under examination (Fig. 12), many of the boundary reflections are expected to reach sensor 5 about $23 \mu\text{s}$ after the arrival of the S_0

Table 4
Simulation summary for simulated experience database creation.

Item	#
Crack positions	43
Crack lengths	12
Actuators	6
Modes	2
Total	6192

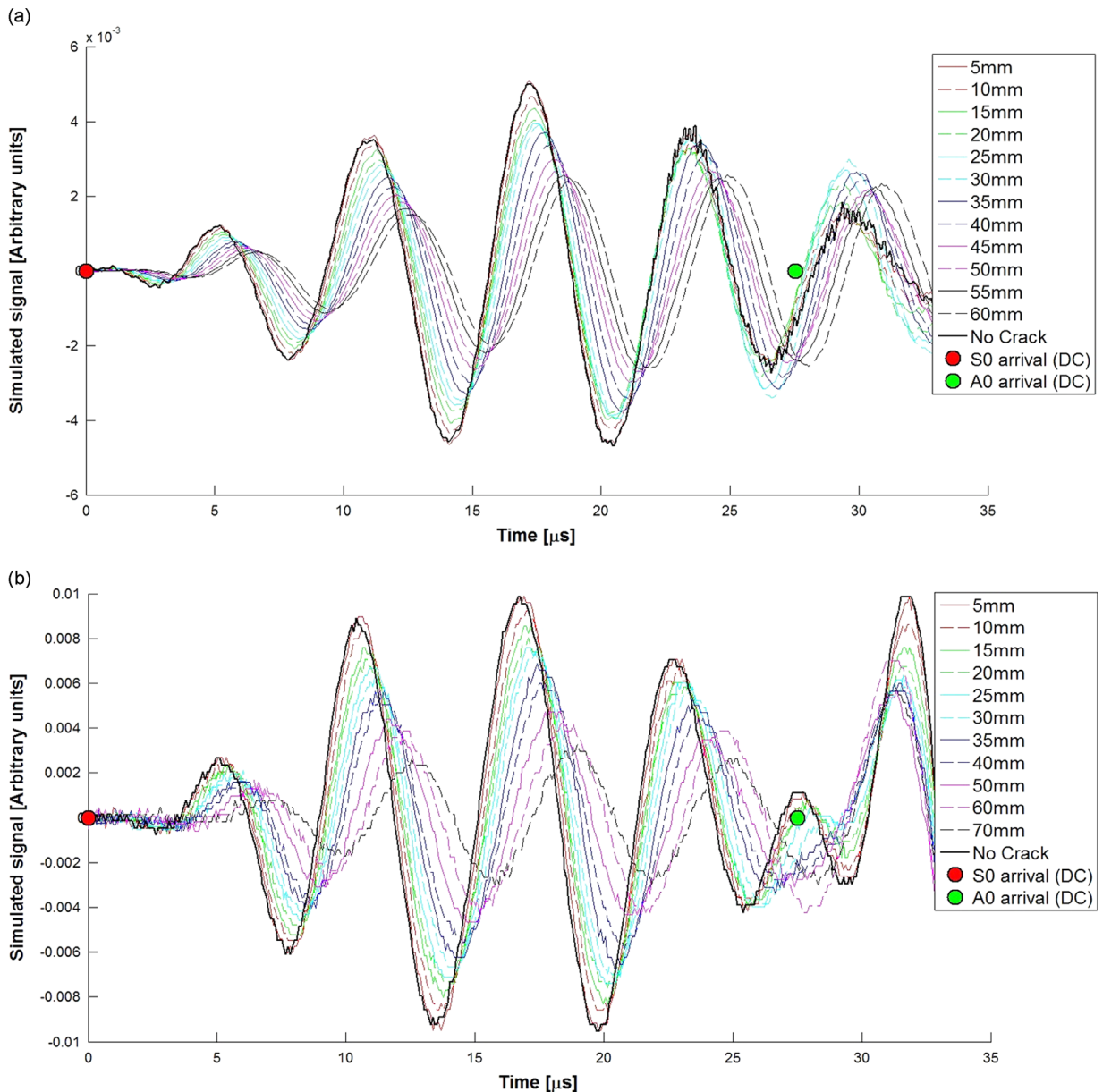


Fig. 11. Crack length effect over the numerical (a) and experimental (b) signals in correspondence of the first S_0 mode wave packet arrival, calculated with analytical dispersion curves (DC).

first wave packet (calculated from the dispersion curves with 5632 mm/ μ s as the S_0 velocity). This explains well the signal complexity observed in Fig. 11 after 23 μ s, which also highlights some model inaccuracies.

The same considerations arise when focusing on the A_0 mode (Fig. 13), made even worse this time by the fact that the A_0 time window is also affected by the presence of S_0 wave packets. This is reflected in a higher discrepancy between the simulated and experimental signals, and this is less controllable in comparison to the S_0 case described above. Nevertheless, a clear effect of damage extent (crack length) on the signal is visible and can be exploited for damage diagnosis.

5.3.2. Damage index characterisation

The information about the damage contained in the signals presented in Figs. 11 and 13 needs to be synthesised into one single value, or damage index, to be used in order to build up a damage index map, where each signal corresponding to an actuator/sensor path will be translated into a damage index.

The *cross correlation* has been selected as a means to evaluate the correlation of a time signal ($y(t)$, relative to any possible damaged case) with the baseline signal ($b(t)$, relative to the undamaged situation). The cross correlation r , as a function of the delay d between the two signals can be expressed as

$$r(d) = \frac{\sum_i [(y(i) - \mu_y)(b(i-d) - \mu_b)]}{\sqrt{\sum_i (y(i) - \mu_y)^2}} \sqrt{\sum_i (b(i-d) - \mu_b)^2} \quad (4)$$

μ_y and μ_b are the mean of the $y(t)$ and $b(t)$ signals respectively, i is an index referring to the i th sample of the signals and $r(d)$ is a vector indicating the correlation of the two signals as a function of the imposed delay d .

In particular, the peak value of the cross correlation vector $r(d)$ was considered as the damage index. One could also use the delay associated to the peak value of $r(d)$ as an index of the damage level, as it is associated with the actual delay induced to the signal by the presence of a damage (as seen in Figs. 11 and 13). However, the former parameter showed better performance for the case under description and will be used from now on.

In order to have comparable damage indexes among the different actuator/sensor paths, the peak of the cross correlation vector was normalised with respect to the peak of the *autocorrelation* function calculated on the baseline signal, for each path and according to the following procedure:

- (1) Calculate the cross correlation function between any signal in the presence of a damage and the related baseline signal (undamaged situation).
- (2) Extract the peak value of the cross correlation function for each available damage case.
- (3) Calculate the autocorrelation function relative to the baseline signal.
- (4) Extract the peak value of the autocorrelation function.
- (5) Normalise the cross correlation peak value obtained at step (2) by the autocorrelation peak value calculated at step (4).
- (6) Repeat for each available actuator/sensor path.

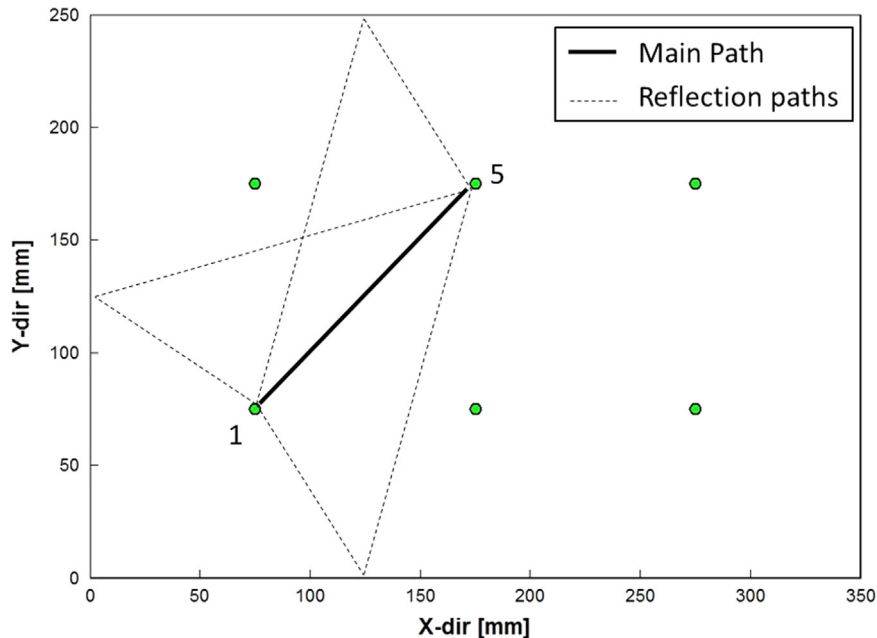


Fig. 12. Scheme of reflections influencing the first wave packet of S_0 mode.

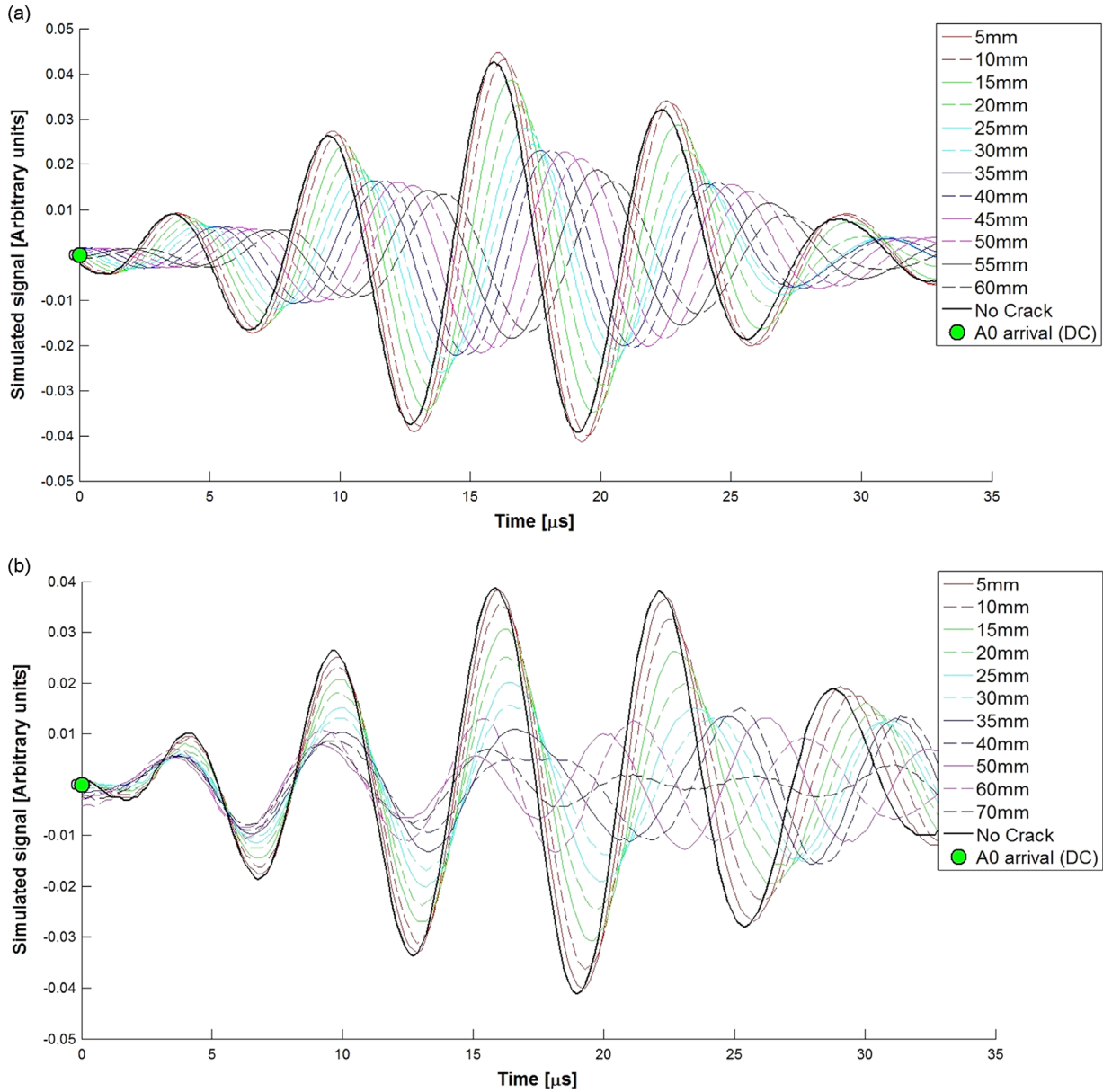


Fig. 13. Crack length effect over the numerical (a) and experimental (b) signals in correspondence of the first A_0 mode wave packet arrival, calculated with analytical dispersion curves (DC).

The same procedure is applied to both the simulated and experimental signals, leading to the definition of numerical and experimental damage index maps. The former are used as numerical experience for the diagnostic algorithm training, while the latter are processed inside the trained algorithm to demonstrate the methodology feasibility.

A comparison of the damage index sensitivity to crack length is provided in Figs. 14 and 15 for the S_0 and A_0 Lamb wave modes respectively. The damage configuration adopted during the experiment (Fig. 4) is considered here for the last model validation step.

Concerning the analysis of the S_0 mode (Fig. 14), a very good agreement is found between the numerical and experimental damage indices. In particular, path 1-5 is expected to be the most sensitive one for the damage under examination and this is reflected in the steep curve in Fig. 14a. On the other hand, path 2-6 is correctly shown to be uninfluenced (Fig. 14b). It should be noticed that, while the numerical experience is provided up to a 60 mm crack length, the experimental crack growth process ended at 70 mm, thus requiring some extrapolation capability of the processing algorithm that will be described in the next section. In addition, the crack centre moved by 5 mm in the x-direction during the last experimental increment of the crack length (refer to Table 1); this is reflected in a sudden decrease of damage index relative to paths 5-2 (Fig. 14c) and 1-6 (Fig. 14d).

Finally, a global comparison has been performed between numerical and experimental damage indices considering all the available signal paths at all the crack length levels, however including one single damage position configuration as highlighted in Fig. 10. The average percentage error is equal to 6 per cent. As a conclusion, the numerically simulated information contained inside the first wave packet of the S_0 propagation mode can clearly be exploited for damage monitoring.

For the sake of completeness, the damage indices calculated for the A_0 mode first wave packet are reported in Fig. 15. Again, the A_0 model is able to reproduce the correct tendency, though with less accuracy with respect to the S_0 model, due to the reasons expressed in previous sections.

6. The diagnostic algorithm

A machine learning approach based on Artificial Neural Networks (ANNs) is used in this work to provide estimates of damage extent and localisation. In particular, the ANN used here is the Multi-Layer Perceptron (MLP). The MLP is one of the most commonly used neural network architectures and uses a multi-layer feed-forward structure. The MLP consists of a collection of connected nodes, namely the input layer nodes, hidden layer nodes and finally the output nodes. In this present work, the input nodes correspond to the damage indices data sets, one hidden layer is used for computation and the outputs are the prediction of the coordinates of the damage location in the x - and y directions and the estimation of crack length for the localisation and quantification ANNs respectively.

The initiation of the above network architecture begins when a set of signal values (damage indices) passes into the input layers, progresses forward through the hidden layer, and the results (predicted damage locations and length) finally emerge at the output layer (prediction of the damage locations). Each node i is connected to each node j in the preceding and the following layers through a connection of weight w_{ij} . Signals pass through each node as follows: in layer k (hidden layer) a weighted sum is performed at each node i of all the signals $x_j^{(k-1)}$ from the preceding layer $k-1$, giving the excitation $z_i^{(k)}$ of the node; this is the passed through a nonlinear *activation function* f to emerge as the output node $x_i^{(k)}$ to the next layer,

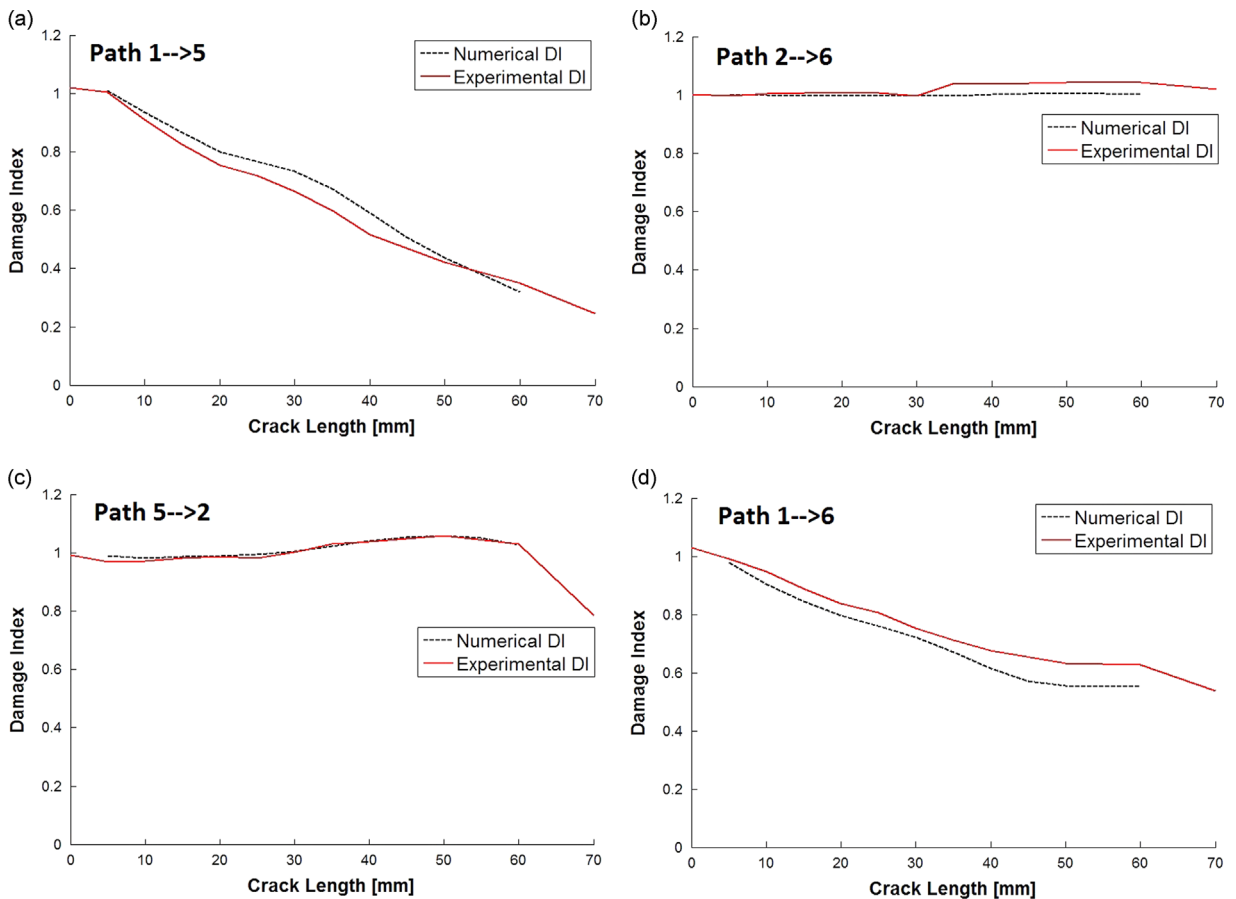


Fig. 14. Damage index sensitivity to crack length, based on the first S_0 wave packet. Comparison of damage index evaluated on numerical and experimental signals, relatively to four selected actuator/sensor paths.

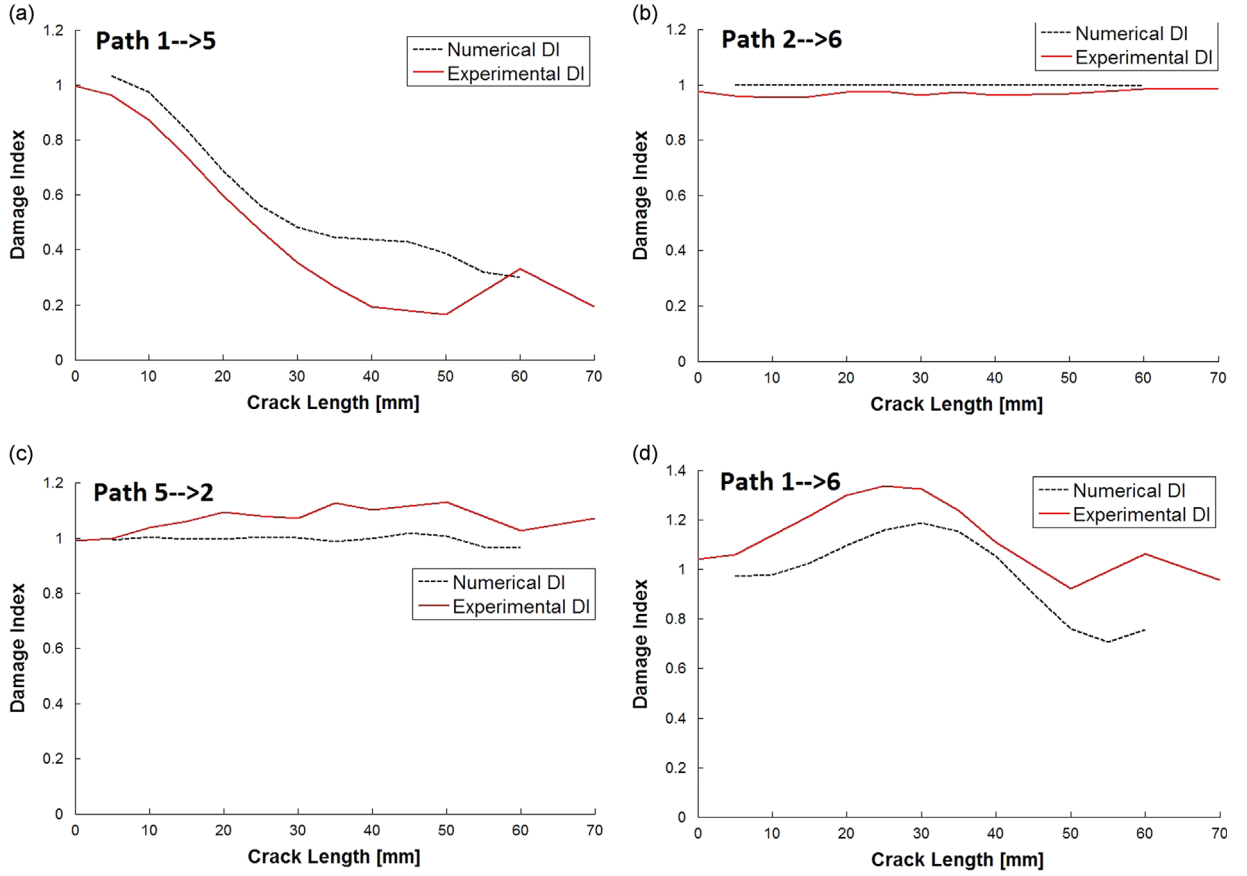


Fig. 15. Damage index sensitivity to crack length, based on the first A_0 wave packet. Comparison of damage index evaluated on numerical and experimental signals, relatively to four selected actuator/sensor paths.

which is highlighted in the following equation:

$$x_i^{(k)} = f(z_i^{(k)}) = f \left(\sum_j w_{ij}^{(k)} x_j^{(k-1)} \right) \quad (5)$$

The activation function f in this case is restricted to $f(x) = \tanh(x)$. One node of the network, the bias node b , is special in that it is connected to all other nodes in the hidden and output layers; the output of the bias node is held fixed throughout in order to allow constant offsets in the excitations z_i of each node. Thus Eq. (5) is transformed into

$$x_i^{(k)} = f(z_i^{(k)}) = f \left(\sum_j w_{ij}^{(k)} x_j^{(k-1)} + b_j^{(k-1)} \right) \quad (6)$$

Finally, the output-unit activation function will generate the output values y_k . Here the linear activation function should be considered, as crack localisation and quantification are posed as a regression problem in this work [2]:

$$y_k = x_j^{(k)} \quad (7)$$

The first stage of using a network to model an input–output system is to establish the appropriate values for the connection weights w_{ij} . This is the *training* or learning phase. The type of training adopted here is a form of supervised learning and makes use of a set of network inputs for which the desired network outputs are known. At each training step, a set of inputs is passed forward through the network yielding trial outputs that can be compared with the desired outputs. If the comparison error is considered small enough, the weights are not adjusted. If, however, a significant error is obtained, the error is passed backwards through the net and a training algorithm uses the error to adjust the connection weights. The algorithm used in this work is the *Scaled Conjugate Gradient* algorithm [22], which is an iterative process to minimise a cost function and consists of two fundamental steps: (i) computation of a search direction calculating the gradient of the cost function at a new point of the parameter space and (ii) line minimisation assuring the new search direction is downhill. Once the comparison error is reduced to an acceptable level over the whole training set, the training phase ends and the

network is established. The networks used for this study were designed and trained using NETLAB [22,23] functions, based on MATLAB code.

Given that the simulated experience database has been validated (Section 5.3) and a damage index has been selected to synthesise the information contained inside each signal path, this knowledge is used here to train the ANN algorithm. The trained algorithm is required to predict the position and the length of a crack when experimental data are processed, having established that the damage index sensitivity calculated on the simulated signals used for training is compatible with that observed on experimental signals. However, to guarantee a sufficient capability of the algorithm trained on numerical data to generalise well on experimental signals is a non-trivial task. The aim of the present section is to define a procedure suitable for the optimisation of the ANN structure and the evaluation of its performances in terms of damage diagnosis. In particular, the following steps will be described in detail:

- (i) Reduction of input space dimension through *Principal Component Analysis* (PCA) [22], Section 6.1.
- (ii) Definition of a *committee* [24,25] of ANN models for damage diagnosis and ANN structure optimisation, in terms of hidden nodes number, based on the *analysis of variance* (ANOVA) [26], Section 6.2.
- (iii) Verification of network performance on simulated data and evaluation of sensitivity to input variations, through *interval arithmetic*, inspired by *information gap theory* [27], Section 6.3.
- (iv) Validation of algorithm capability for generalisation on a real experimental test, separately presented in Section 7.

6.1. Reduction of input space dimension

In general terms, a diagnostic system based on Lamb wave propagation and constituted by N transducers allows the definition of $N(N-1)$ paths. As previously described, each path is related to a damage index and corresponds to an input node in the ANN structure. Specifically referring to the problem under examination (6 piezoelectric transducers), this means having 30 input nodes, corresponding to 30 damage indices, relative to 30 paths. Though this level of complexity is affordable for the diagnostic system herein considered, one might be interested in reducing the input space dimension if a denser sensor network is installed. This can be accomplished through *Principal Component Analysis* (PCA), which is one of the most commonly used feature extraction techniques. It has been chosen because it is easy and fast to compute and it retains maximal information amongst all linear projections [22]. Due to the widespread knowledge of the method, only its application will be described hereafter.

516 damage cases have been simulated, combining 43 damage centre positions with 12 crack lengths. Each damage case is characterised by its own damage index map, constituted by 30 damage indices calculated on the 30 available paths (refer to Table 2). This generates a 30×516 original input matrix containing the global simulated experience. After calculating the covariance matrix of the input database, its eigenvectors are the principal components (PC) and their associated eigenvalues λ_i are an index of the amount of variance retained by any i th component. It is usual to list the PCs in descending order of eigenvalues, so that the first PC corresponds to the largest variance of any linear combination of the original variables. A useful heuristic method to reduce input space dimension is to plot singular values (Fig. 16) to see if there is a point at which the values level off, or to select the first M components so that a fraction of the total variance will be retained [22], calculated as follows:

$$\frac{\sum_{i=1}^M \lambda_i}{\sum_{i=1}^d \lambda_i} \quad (8)$$

where d and M are the original and the reduced input space dimension respectively. In particular, the new input matrix is obtained by multiplication of the original input matrix by the matrix containing the first M eigenvectors.

30 eigenvalues have been calculated based on the available numerical database and have been reported in Fig. 16, together with the cumulative distribution calculated as in Eq. (8). It appears that the first 15 PCs retain nearly 95 per cent of the total variance; as a matter of fact, the other 15 PCs have an almost null eigenvalue. This result is related to the fact that the path between two transducers is being considered twice: as an example, the signal simulated from actuator ID1 to sensor ID5 is the same as from ID5 to ID1, apart from some discrepancies due to imperfect exclusion of boundary reflections inside the considered time window.

6.2. Optimisation of algorithm structure

Two ANN structures have been trained, receiving as input data a 15×516 matrix containing the simulated experience (conveniently “projected” onto its principal components) and as output data the crack position (the localisation ANN) and the damage length (the quantification ANN) respectively.

There would normally be issues of generalisation with such a small training set. The received wisdom in the ANN field demands that there usually be 10 training patterns per weight [24,25] and this condition is not remotely met here. However, this is valid if no regularisation is used, while different techniques have been adopted here to guarantee algorithm generalisation.

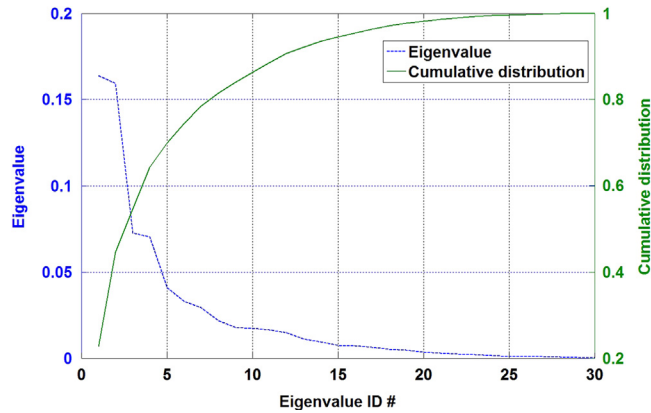


Fig. 16. Eigenvalues calculated on the numerical database and cumulative distribution (eigenvalues are sorted in descending order).

First, *early-stopping* based on a validation set [25] has been used here. In fact, the entire dataset has been randomly split into two subsets, namely training and validation, containing 80 per cent, and 20 per cent of the data respectively.

The use of a *committee* of ANNs was also considered here as an aid to generalisation. While the results obtained with the single ANN are in some cases acceptable, it is often found that improved performance can be obtained by combining multiple models together instead of just using a single model in isolation. It can be demonstrated [24,25] that the error of the committee is lower than the average error associated to a single ANN by a factor equal to the number of replicated versions of the initial model. This is true under the hypothesis that the errors due to the individual models are uncorrelated. Otherwise, the error reduction is smaller, but it can be demonstrated that the expected committee error will not exceed the expected error of the constituent models.

Moreover, noise was added in early trials in order to find if this would further improve generalisation, but it was found that the use of early-stopping alone during the training of a committee of ANNs was adequate.

The simplest way to construct a committee is to average the predictions of a set of individual ANN models. This is valid especially if the prediction distribution of the committee can be assumed Gaussian, which will be confirmed later for the current work (Section 7). Different ANN models can be created either by modifying the ANN structure or by maintaining the ANN design unchanged but training it with different datasets. As the ANN structure here will be optimised in terms of hidden neuron number, the second option will be adopted in the following. However, in practice, only one single dataset is available (the numerical database) and a way to introduce variability between the different models within the committee has to be found. One way is to use *bootstrap* datasets [24,25], that is to say to create new datasets by sampling from the initial available domain. Each ANN belonging to the committee will be trained and validated with a different dataset, randomly selecting those test cases that will fall into the training and validation sets, however using all the data available inside the initial domain. This technique has been adopted here to train 60 ANNs for crack length quantification and 60 ANNs for crack centre position estimation. Each ANN will have the same structure, optimised as described in the following, but it is trained and validated with a different dataset.

Defining the optimal structure of an ANN is a complex procedure, especially because of the number of variables involved during training, often correlated with each other. In a MLP, the main parameters to be defined are the number of hidden layers and the number of nodes belonging to each hidden layer. Intuition suggests that too low and too many nodes will induce poor and over-fitting of the training data respectively [25]: an optimal condition appears to exist. Nevertheless, training duration is another important aspect to be considered: again, excessive training might induce over-fitting of the database and an early stopping procedure (based on a validation set) can be used to decide whether to continue or not the training procedure. It might happen that the adoption of early stopping would guarantee fairly good performances (e.g. in terms on Root Mean Square Error – RMSE) even if a relatively more complex structure is allowed. This is well reflected in Fig. 17, where the validation RMSE distributions (build through the output of 60 ANNs for damage localisation and quantification) appears to level off after a certain number of hidden nodes. A distribution of RMSE is available for each network configuration because repeated training with different training and validation subsets and different weight initialisation can generate slightly different models. A rigorous method for the identification of the best performing ANN structure is required and using committees of ANNs during model structure optimisation would allow a statistical selection of the ANN structure parameter(s), which is useful especially in those applications where a significant model variability is found. It is clear from Fig. 17 that the effect of the hidden node number over the performance of the networks is significant only up to a certain limit. The problem at issue becomes the identification of this limit and the *analysis of variance* (ANOVA) can be used in order to statistically select the number of nodes inside one single hidden layer.¹

¹ Some early trials with a different number of hidden layers were made to find if this would improve the algorithm performance. Nevertheless, no significant improvement was found and one single hidden layer will be considered from now on.

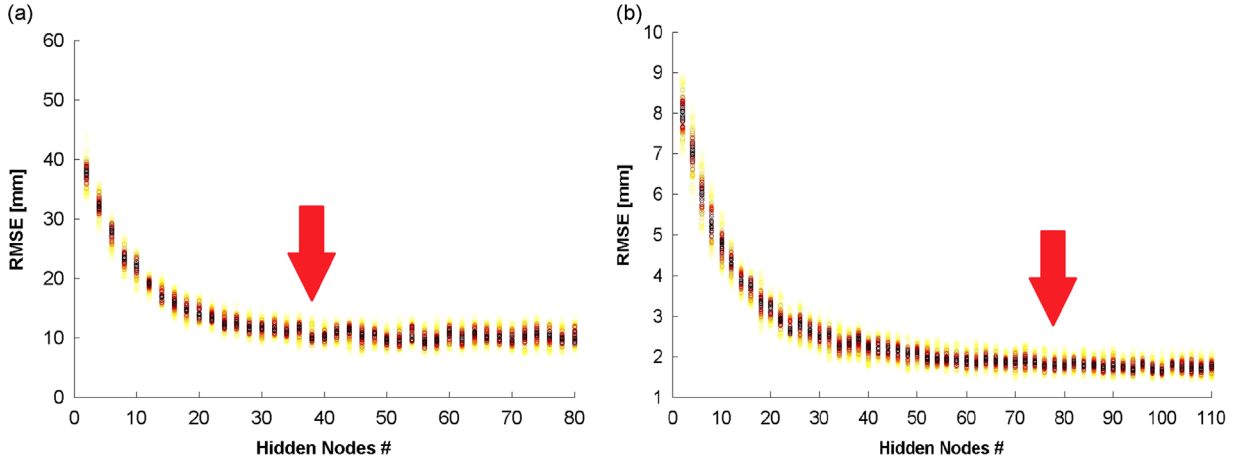


Fig. 17. Optimisation of node number inside the hidden layer. (a) RMSE distributions as a function of hidden node number for the localisation ANN. (b) RMSE distributions as a function of hidden node number for the quantification ANN.

ANOVA is a statistical methodology typically adopted in the design of experiments framework. It can be used to verify the influence of one or more *factors* over the observed variable, provided the observations relative to different levels (or *treatments*) of the same factor are available. Suppose a linear statistical model describes the observations from a single factor experiment, like the following:

$$Y_{ij} = \mu + \tau_i + \varepsilon_{ij} \begin{cases} i = 1, 2, \dots, a \\ j = 1, 2, \dots, n \end{cases} \quad (9)$$

where Y_{ij} is a random variable indicating the j th observation for the i th treatment, μ is a parameter common to all the treatments (often referred to as *grand mean*), τ_i is a parameter associated to the i th treatment and indicating the effect of that specific treatment, ε_{ij} is a random error component, a and n are the number of treatments and observations per treatment respectively. For the purposes of the present study, ANOVA consists in verifying the equality of the a means associated to the a treatments. This translates into a hypothesis test as follows:

$$\begin{aligned} H_0: \tau_1 = \tau_2 = \dots = \tau_a = 0 \\ H_1: \exists i \mid \tau_i \neq 0 \end{aligned} \quad (10)$$

Rejecting H_0 implies there is sufficient evidence in the observations of the influence of the considered factor on the observation mean. The assessment is made based on a *Fisher* distribution test statistic and assuming a limit on the *First Type error*,² also referred to as *significance level* for the test. A deeper treatment of the theory behind ANOVA is outside the scope of the present paper; nevertheless the interested reader might refer to [26] for a detailed explanation of the theory and the hypothesis requirements within ANOVA.

Specifically referring to the case under examination, ANOVA has been recursively applied in order to highlight the influence of the *factor* “hidden node number” over the RMSE parameter. In particular, the higher level (*treatment*) of the factor is considered fixed to 80 and 110 hidden nodes for the localisation and quantification problems respectively. Then, the lower level (*treatment*) of the factor is increased at each iteration from 2 to 78 (for the localisation problem) and from 2 to 108 (for the quantification problem). The limit number of hidden layer nodes for H_0 rejection have been indicated in Fig. 17a and b for the localisation and quantification problems respectively. 5 per cent probability of first type error has been assumed as significance level for the test, as it is typically found in literature [26]. Focussing on *localisation* (Fig. 17a), sufficient evidence of factor influence over the considered parameter (here RMSE) was found when the factor levels between 2 and 38 were taken into consideration inside the test. In the same way, there is no evidence of factor influence over the *quantification* ANN performance when the hidden node levels from 80 to 110 are considered (Fig. 17b). As a conclusion, 38 and 78 hidden layer nodes have been chosen as optimal for the localisation and quantification ANN committees respectively.

A summary of the parameters used for the ANN design is finally provided in Table 5.

6.3. Verification of network performance on simulated data

Especially when simulated data are used for training, it is important to appreciate the sensitivity of the algorithm to variation of input data and many methods are available to solve this problem. One traditional approach would be to employ

² Probability to reject H_0 when H_0 is true (false positive).

Monte-Carlo techniques involving randomising the input data (within defined bounds) a large number of times, and evaluating the change in output error for each discrete set of inputs. This technique has a significant drawback, especially when applied to nonlinear MLP networks, in that it is impossible to be sure of mapping all possible combinations of variation in input space to output space unless an unfeasibly large number of sample points are used. Since interest lies in understanding the worst possible performance of the classifier in the presence of input data uncertainty, a Monte-Carlo approach would therefore not provide certainty of behaviour under all possible input data conditions. It was to circumvent this problem that a non-probabilistic interval number approach to the input uncertainty was adopted [27]. Interval numbers occupy a bounded range along the number line, and can be defined as an ordered pair of real numbers $[a, b]$ with $a < b$ such that

$$[a, b] = \{x | a \leq x \leq b\} \quad (11)$$

Interval numbers have specific rules for the standard arithmetic operations of addition, subtraction, multiplication and division, and their forward propagation through ANNs is calculable provided that the transfer functions associated with the networks are monotonic (the tanh function used here as activation function satisfies this condition). The interested reader can refer to the dedicated literature for a more detailed explanation of interval arithmetic and information gap theory as the aim here is only to describe its application to the problem under investigation, specifically the uncertainty propagation for the evaluation of the algorithm sensitivity to input variations.

In practice, each input value x_i of the dataset was intervalised by a parameter α such that

$$[x_{ia}, x_{ib}] = [x_i(1 - \alpha), x_i(1 + \alpha)] \quad (12)$$

The choice of α is a critical aspect to be considered. While probabilistic methods suffer from the common drawback that probability distributions (usually described through their low-order moments) are not sufficiently validated at their extremes, for interval-based techniques (such as the one adopted here) the problem translates in a reliable assumption of the uncertainty level. The α parameter has been set to 0.06, indicating a ± 6 percent variation with respect to the original input data. It has been calculated in Section 5.3.2 and it is an index of the extent of experimental mismatches with respect to the reference numerical information used for algorithm training. In real applications, this value is usually not available and one should refer to literature or to in-house modelling experience in order to guess it. It is also important to consider that only the uncertainty level associated to numerical modelling has been included here into α , as no significant environmental variations were considered during the experimental test (e.g. the effect of load, temperature, humidity, etc.). In practice, the intention here is to propagate only numerical uncertainty through the network. Nevertheless, in the most general application, one has to consider a larger interval taking into account for any mismatch with respect to the baseline reference case.

The interval was then forward propagated through the networks under test and the interval outputs $[y_a, y_b]$ have been calculated, to be compared with the target values. Differently from [27], where interval propagation through a single ANN was considered, in this work the uncertainty is propagated through a committee of networks. In practice, being M the number of models inside the committee, M output intervals are obtained and the problem at issue is how to combine these intervals to provide useful indication of network output robustness.

The first method consists in taking as output interval the maximum and minimum limits among those available from the M models, as follows:

$$[o_a, o_b] = [\min_k(o_{ka}), \max_k(o_{kb})] \quad (13)$$

where o_a and o_b are the upper and lower *absolute interval boundaries* for the output, o_{ka} and o_{kb} are the upper and lower output interval boundaries for the k th model belonging to the committee. This output interval will contain information on both the output dispersion of the 60 ANNs inside the committee and the output interval associated to each one. This can be interpreted as an index of the robustness of the single ANN, including the sensitivity to input and training uncertainties.

The second method leverages on the fact that the real committee output of interest is the average of the 60 ANN outputs. Thus, it is also interesting to evaluate the output interval relative to the ensemble average committee. Though averaging procedures might be in contrast with a non-probabilistic approach based on interval propagation, it is reasonable to assume the average calculation as a part of the ensemble diagnostic model itself, then propagating the single output intervals through it. The interval boundaries of the committee output consist of an average of the interval boundaries calculated for

Table 5
Parameters for Artificial Neural Network design.

Diagnostic Level	ANN type	Input layer	Output layer	Hidden layer nodes	Training strategy	ANN # in committee
Localisation	Function fitting	15 Damage indices	[X,Y] Position of crack centre	38	Scaled conjugate gradient	60
Quantification	Function fitting	15 Damage indices	Crack length	78	Scaled conjugate gradient	60

the single ANNs belonging to the committee, as follows:

$$[\bar{o}_a, \bar{o}_b] = \left[\sum_{k=1}^M o_{ka}/M, \sum_{k=1}^M o_{kb}/M \right] \quad (14)$$

where \bar{o}_a and \bar{o}_b are the upper and lower *averaged interval boundaries* for the output. It must be underlined that the resulting output interval boundaries will be an index of sensitivity to input variation for the committee output and not for the single ANN belonging to the committee. For this reason, narrower boundaries are expected in this case.

Three examples of diagnostic output are provided in Fig. 18 where numerical signals relative to three simulated crack propagations are processed forward through the algorithm. The optimised network configuration described above is used hereafter. Localisation output is presented in the left side subplots while the quantification output for the same damages is on the right side.

Focussing on localisation, a large error of the committee output is found for smaller cracks, while the output converges toward the target crack centre position during crack propagation. Concerning the interval outputs, three phases can be distinguished: (i) relatively big gaps are obtained for smaller cracks, indicating the difficulty in interpreting the signals, due to the very small sensitivity to damage; (ii) the output gap is narrowed while the crack length increases demonstrating the robustness of network training and (iii) a final enlargement of the interval boundaries is found for longer cracks, indicating a more complex behaviour of damage indices. Considering only those cracks from 20 mm to 60 mm long inside the numerical database, the Mean Square Error of the committee output was found to be 0.13 per cent of the skin area within the sensor grid (100 mm × 200 mm). If the absolute output intervals are considered, the average worst case error for localisation (WC_l) is 21 per cent of the skin area within the sensor grid (A), considering a ± 6 percent variation with respect to the original input data and calculated as

$$WC_l = \frac{100}{NA} \sum_{n=1}^N \max [(o_{x,n,a} - t_{x,n})^2, (o_{x,n,b} - t_{x,n})^2] + \max [(o_{y,n,a} - t_{y,n})^2, (o_{y,n,b} - t_{y,n})^2] \quad (15)$$

where N is the total number of input cases belonging to the average, $t_{x,n}$ and $t_{y,n}$ are the target x and y coordinates of crack centre for the n th case and $[o_{x,n,a}, o_{x,n,b}]$ and $[o_{y,n,a}, o_{y,n,b}]$ are the boundaries of the absolute output interval for the x and y coordinates respectively (for the n th case). This result is representative of the single network output sensitivity to input variation and training uncertainty. If the same calculation is repeated for the averaged interval boundaries (substituting $[\bar{o}_{x,n,a}, \bar{o}_{x,n,b}]$ and $[\bar{o}_{y,n,a}, \bar{o}_{y,n,b}]$ to $[o_{x,n,a}, o_{x,n,b}]$ and $[o_{y,n,a}, o_{y,n,b}]$ respectively), the average worst case error for localisation (WC_l) becomes 1.5 per cent of the skin area within the sensor grid (A), which is indicative of the committee output sensitivity to input uncertainty.

Focussing on quantification, the committee output error is very small, indicating that a clear trend of input data was identified and learnt during training. In particular, the RMSE associated to the committee output is 0.58 mm, corresponding to less than 1 per cent of the longer crack assumed during training (60 mm). Interval outputs tends to enlarge as a function of target crack length, due to the more complicated behaviour of damage indices for longer cracks. If the absolute interval boundaries are considered, the average worst case error for quantification (WC_q) is 19 per cent of the longer crack ($CL_{\max} = 60$ mm) assumed during training (corresponding to 11.6 mm), considering a ± 6 percent variation with respect to the original input data and calculated as

$$WC_q = \frac{100}{NCL_{\max}} \sum_{n=1}^N \max [\text{abs}(o_{n,a} - t_n), \text{abs}(o_{n,b} - t_n)] \quad (16)$$

where N is the total number of input cases belonging to the average, t_n is the target crack length for the n th case and $[o_{n,a}, o_{n,b}]$ are the lower and upper limits of the crack length absolute output interval respectively. If the same calculation is repeated for the averaged interval boundaries (substituting $[\bar{o}_{n,a}, \bar{o}_{n,b}]$ to $[o_{n,a}, o_{n,b}]$), the average worst case error for quantification (WC_q) is 12 per cent of the longer crack ($CL_{\max} = 60$ mm) assumed during training (corresponding to 7.5 mm). This confirms the advantages due to the application of a committee of ANNs.

7. Test results

The Lamb wave signals have been acquired during the experiments for each available actuator/sensor path (path map) and for different levels of artificial crack length (refer to Table 1). In particular, the experimental activity attempts to replicate the same damage evolution as described in Figs. 18a and b. The data were first processed by an algorithm which extracted the damage index defined in Section 5.3.2, thus creating a damage index map. The experimental damage index map was then processed through the algorithms defined in Section 6: first the input space has been reduced through PCA and then projected input signals have been processed forward into the ANN committees, which were trained with numerical experience. The obtained results are presented in Figs. 19 and 21 respectively for crack length quantification and damage centre localisation.

Focusing first on damage quantification (Fig. 19), the real and the estimated crack lengths are presented as a function of the acquisition steps. The uncertainty related to the adoption of a single model is clear when one considers the distribution of the outputs from the 60 committee ANNs trained for crack quantification. It can be noticed how the average value of this

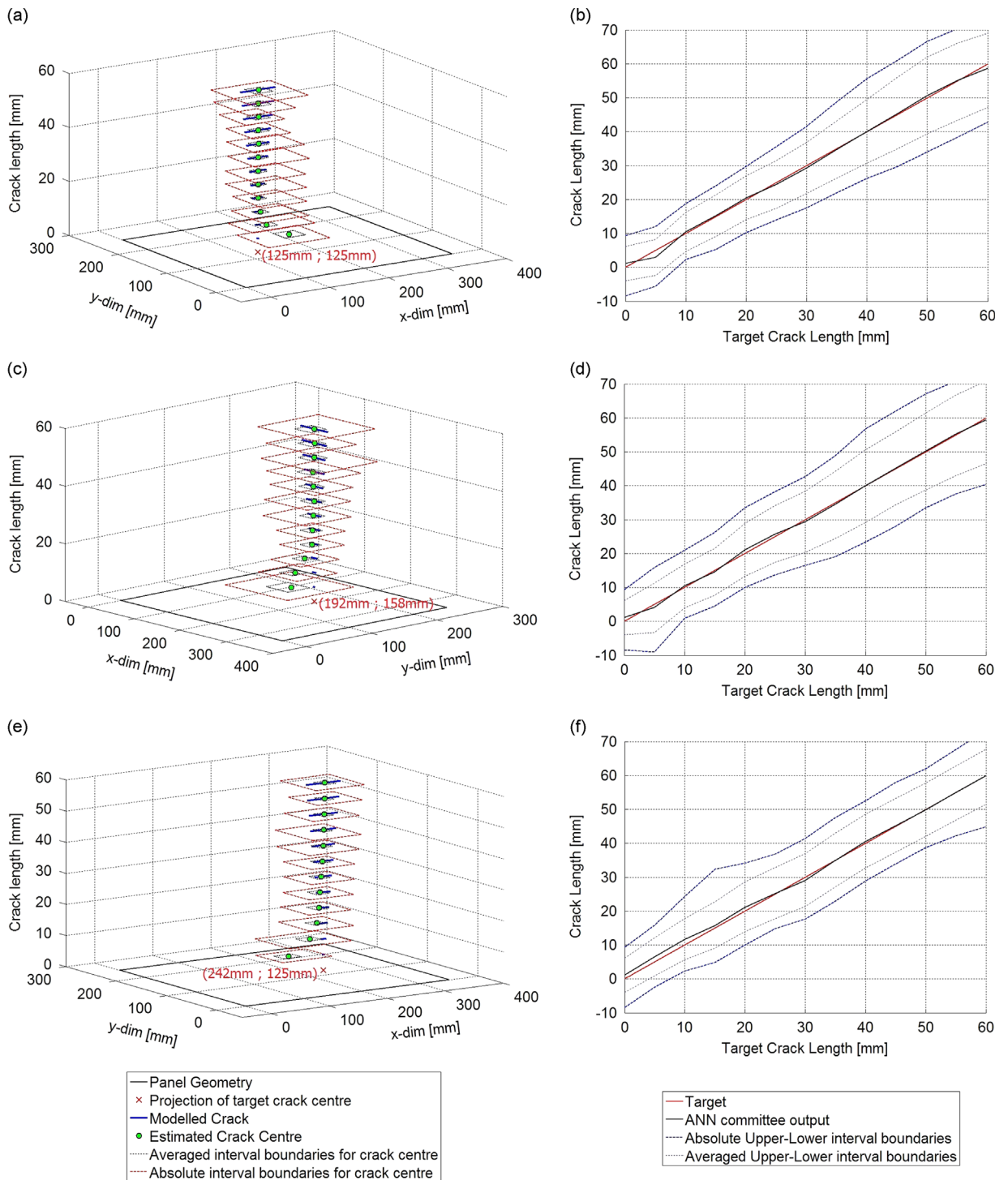


Fig. 18. Evaluation of diagnostic performance for 3 simulated crack propagations with different crack centres. On the left, estimation of crack centre during simulated crack propagation. On the right, estimation of crack length during simulated crack propagation (relative to crack centres on the left figures). Boundaries calculated with information gap theory are provided for both localisation and quantification of the crack.

distributions is particularly effective in describing the real crack length. The error level between the committee output and reality remains more or less constant during the acquisition steps (Table 6) as the estimation oscillates around real crack length. On the other hand, the variance of the committee distribution increases for longer cracks. In fact, longer cracks affect the signal of a larger quantity of paths, thus increasing the probability of making errors in the diagnosis. The adoption of a

network committee appears to be particularly effective to tackle this problem. Although ANNs should not be used for extrapolation due to the unknown behaviour of the network function outside the training boundaries, no error increase was recorded when the experimental signals relative to a 70 mm long crack were processed, even though the numerical experience was only provided for up to 60 mm long cracks. Finally, the output intervals calculated in Section 6.3 have also been reported: the numerically predicted absolute interval boundaries are representative of the output uncertainty for the single network when experimental data are processed through the algorithm; also the committee outputs calculated from the experimental signals always fall within the two averaged numerical interval boundaries previously calculated in Section 6.3.

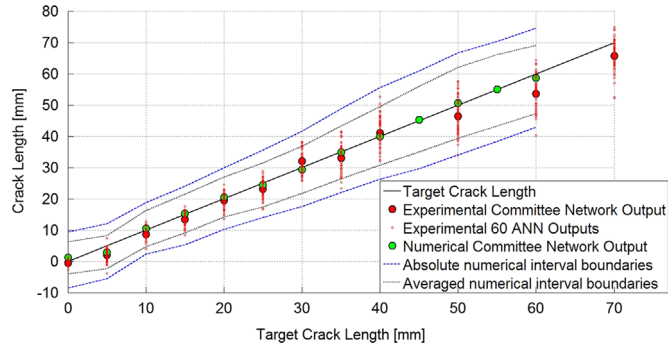


Fig. 19. Crack length quantification results for different experimental acquisition steps; the output of each single ANN is provided, together with the result coming from their committee; the real crack length is also shown as a term of comparison.

Table 6

Errors (absolute value) for crack length quantification and damage localisation as a function of real damage length.

Crack length [mm]	0	5	10	15	20	25	30	35	40	50	60	70	Average
Crack length ERR [mm]	0,5	3	1,3	1,5	0,5	1,8	2,1	1,9	1,1	3,6	6,4	4,3	2,3
x-Pos ERR [mm]	\	77,5	17,6	7,9	3,7	4,0	6,1	0,7	0,3	2,6	2,5	2,0	11,4
y-Pos ERR [mm]	\	10,6	0,2	2,1	2,3	3,4	2,4	5,1	5,3	7,5	6,0	2,8	4,4

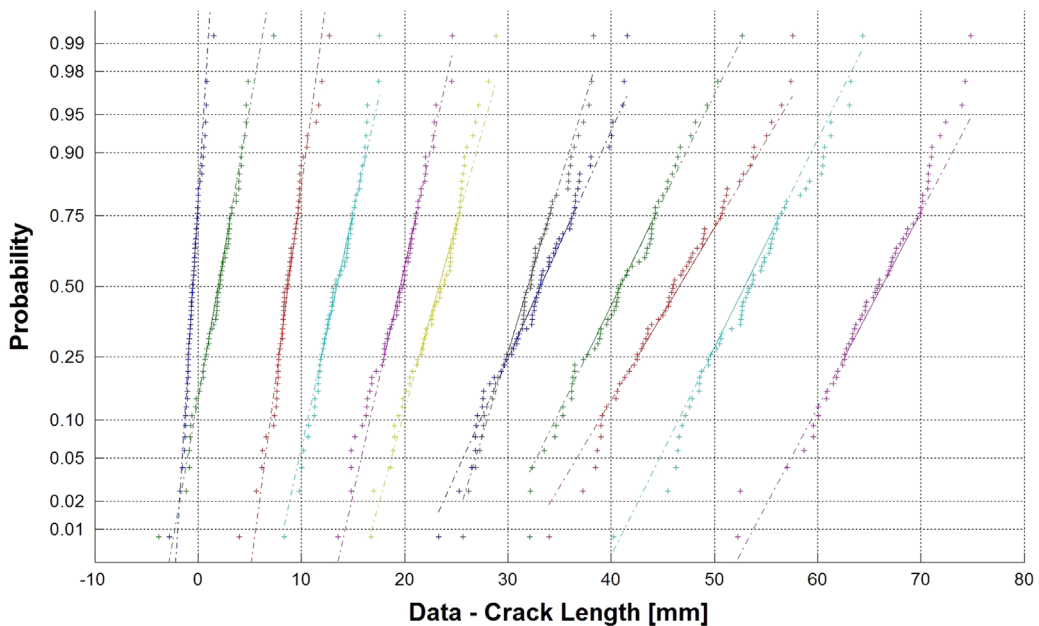


Fig. 20. Normal probability plots separately evaluated for each committee output distribution in Fig. 19. Each curve from the left to the right side refers to a different acquisition step number in Fig. 19.

Furthermore, the same committee output distributions appear to be Gaussian if one refers to Fig. 20, which justify the adoption of the mean as the expected value. The Normal Probability Plot (NPP) is shown in Fig. 20 separately for the output data provided by the algorithm at each acquisition step. Each plot refers to the inference on a different crack length level. The NPP is a graphical technique to assess whether or not a data set is normally distributed. If a sufficient number of samples are extracted from an exact normal distribution, their NPP will be represented by a straight line, which is shown as reference. Any departure from the straight line indicates deviation from normal distribution. It is fairly clear that the committee output distribution can be approximated as Gaussian independently from the measured crack length.

The same considerations are valid for the crack centre localisation (Fig. 21). The distribution of the outputs calculated from the 60 committee ANNs is shown at different crack length levels, together with the target position for the crack centre and the total committee output. A consistent error is recorded for the smallest crack, nevertheless it rapidly decreases to acceptable values as the damage increases, as it is confirmed in Table 6. Again, the variability of the single network output increases as the crack becomes bigger, however it appears to (visibly) maintain a bivariate normal distribution. NPP can be evaluated also for the case of crack centre position inference, nevertheless they have not been reported here for simplicity. However, the average of the distribution always visibly falls in the centre of this distribution cloud. The algorithm capability for extrapolation is confirmed also for the crack centre position inference. Moreover, it is important to note that the crack

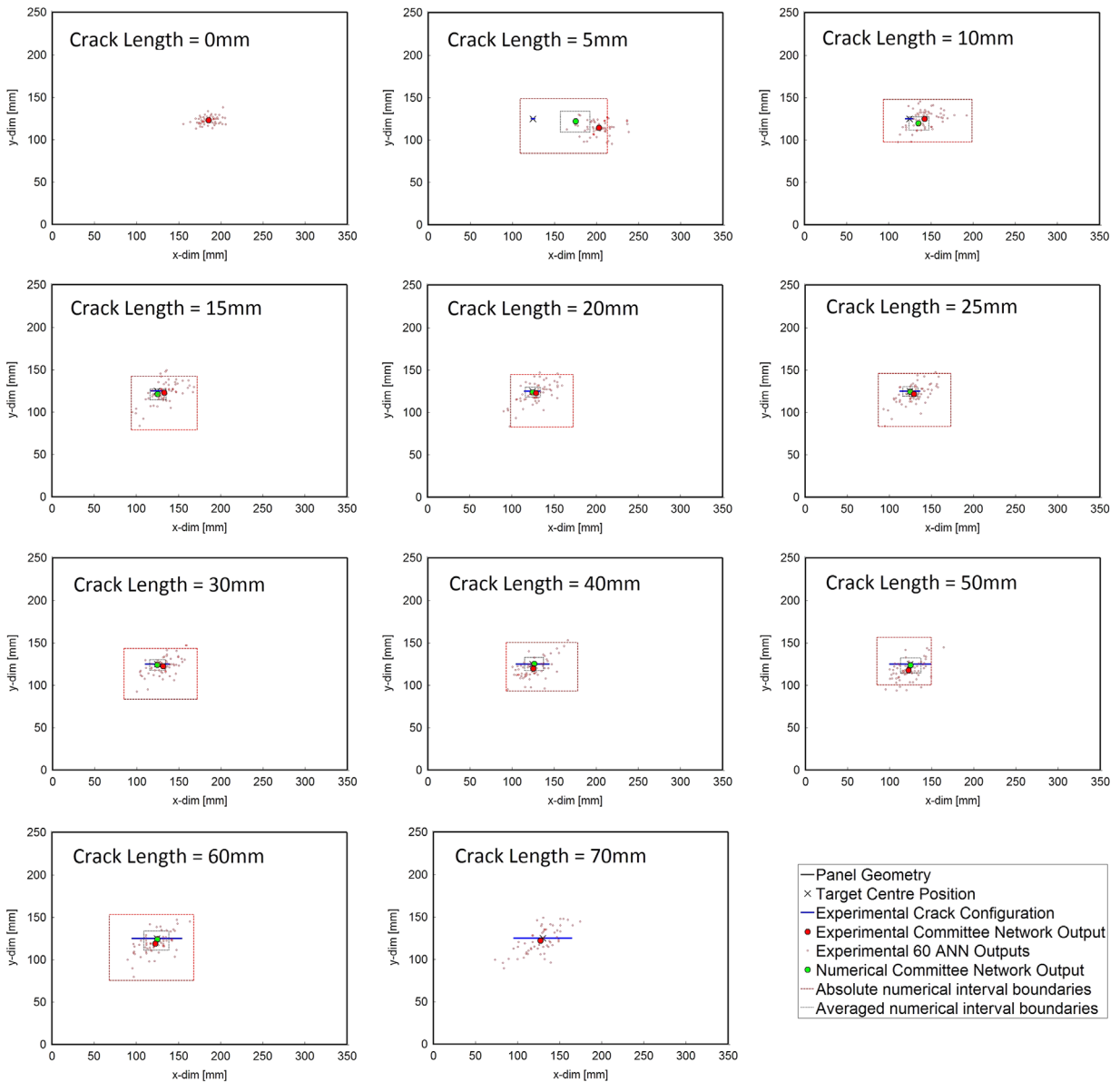


Fig. 21. Crack centre localisation results for different experimental acquisition steps; the output of each single ANN is provided, together with the result coming from their committee; the target crack centre position is also shown as a term of comparison.

centre position moved by 5 mm for the last acquisition during the experiments and that this is reflected in a consequent shift of the algorithm estimate, proving the sensitivity and precision of the proposed methodology. Finally, the output intervals calculated in Section 6.3 have also been shown in Fig. 21: when the crack is longer than 5 mm, the numerically predicted absolute interval boundaries are representative of the output uncertainty for the single network when experimental data are processed through the algorithm; furthermore, the committee outputs calculated from the experimental signals fall within the averaged numerical interval boundaries previously calculated in Section 6.3.

8. Conclusions

A combined data-based and model-based machine learning approach to damage diagnosis is the subject of the present article. The approach takes advantage of physics-based numerical models in order to create the necessary experience (training data) for a neural network algorithm in order to recognise ongoing damage. The proposed methodology proved to be particularly effective for the solution of the damage location and quantification levels typically present in a full SHM algorithm structure.

The main requirements for the success of the methodology are the availability of a baseline signal, relative to the undamaged structure, and a parametric numerical model. A thorough procedure for model verification and validation is a critical aspect treated in detail inside the paper. Once the model is validated on the baseline signal and a robust Lamb wave packet has been selected as a diagnostic feature from the time signals, the model sensitivity to the crack is expected to be compatible with the actual sensitivity on the real structure. The robustness of the selected signal window should be checked with reference to any possible boundary reflections of the selected mode as well as with respect to any disturbance which might come from additional or unexpected Lamb wave modes.

The methodology was tested on an aluminium skin equipped with 6 piezoelectric sensors/actuators. The numerical model for Lamb wave propagation was iteratively run, changing the position and the length of the crack damage, thus simulating a totality of 516 crack cases. This allowed the creation of a numerical experience database, thus avoiding the requirement for pure experimentally based learning, one of the factors that have made SHM implementation unfeasible.

Data pre-processing is required in order to extract the most interesting information from both the numerical and experimental signals, that is to say a damage index. The normalised peak of the cross correlation function, calculated between the baseline signal and the signal in the presence of damage, has been used for the current application, and the resulting 516 damage index maps have been used for algorithm training.

A procedure suitable for the optimisation of the ANN structure and the evaluation of its performances in terms of damage diagnosis has been defined. After reducing the input space dimension through principal component analysis, two algorithm structures have been statistically optimised for localisation and quantification of damage. Analysis of variance has been used to define the number of hidden neurons above which no further network performance improvement is expected. Especially if using numerically simulated signals as a reference information for diagnosis, the verification and maximisation of the algorithm generalisation capability will have to be specifically addressed and various regularisation techniques were also adopted in this study. First, 60 ANN models with the same structure have been grouped into committees in order to improve algorithm generalisation. Then, the committees (one for localisation and one for quantification) have been trained with simulated experience through a scaled conjugate gradient optimisation technique and early stopping procedure. Artificial noise was also added to find if this would further improve generalisation, nevertheless the use of early-stopping alone during the training of a committee of ANNs was sufficient. Finally, a technique inspired by information gap theory and based on interval arithmetic has been adopted to predict the network performance in the presence of input uncertainties, however based on simulated signals.

The same algorithm structure was tested with experimental data, validating the performance of the proposed system. In particular, the average error obtained for crack length estimation was 2.3 mm (the maximum crack length was 70 mm) while the mean error on localisation was 11.4 mm and 4.4 mm, for the x and y coordinates respectively (the panel dimensions were 350×250 mm).

Of course, the biggest limitation in the utilisation of such a methodology is the enormous amount of simulations that need to be run in order to scan large areas. However, on one hand the new technology based on GPUs allows for a dramatic reduction in computational time; on the other hand the adoption of symmetries as well as design experience can also reduce by far the requirements for simulations.

Acknowledgements

The authors would like to acknowledge the entire Dynamic Research Group of Sheffield University for the exchange of valuable comments, in particular Mr. Les Morton and Mr. Kai Yang for their help during experiments, Dr. Sofia Pavlopoulou for her comments on Lamb wave simulation, Mr. Andrew Spencer for his help in LISA SW utilisation and Mr. Nikolaos Devilis for his opinions on Artificial Neural Networks. Finally the authors would like to thank Professor Wieslaw Staszewski and colleagues of AGH University, Karkow for providing the CUDA implementation of the LISA algorithm. The authors acknowledge the use of the software package NETLAB developed by Ian Nabney of Aston University (<http://www.ncrg.aston.ac.uk/netlab/>).

References

- [1] C.R. Farrar, K. Worden, *Structural Health Monitoring: A Machine Learning Perspective*, John Wiley and Sons, 2012.
- [2] F. Mustapha, G. Manson, K. Worden, S.G. Pierce, Damage location in an isotropic plate using a vector of novelty indices, *Mechanical Systems and Signal Processing* 21 (2005) 1885–1906.
- [3] E. Papatheou, G. Manson, R.J. Barthorpe, K. Worden, The use of pseudo-faults for novelty detection in SHM, *Journal of Sound and Vibration* 329 (2010) 2349–2366.
- [4] E. Papatheou, G. Manson, R.J. Barthorpe, K. Worden, The use of pseudo-faults for damage location in SHM: An experimental investigation on a Piper Tomahawk aircraft wing, *Journal of Sound and Vibration* 333 (2014) 971–990.
- [5] J. Kudva, N. Munir, P.W. Tan, Damage detection in smart structures using neural networks and finite-element analysis, *Smart Materials and Structures* 1 (1992) 108–112.
- [6] K. Worden, A.D. Ball, G.R. Tomlinson, Fault location in a framework structure using neural networks, *Smart Materials and Structures* 2 (1993) 189–200.
- [7] C. Sbarufatti, A. Manes, M. Giglio, Performance optimization of a diagnostic system based upon a simulated strain field for fatigue damage characterization, *Mechanical Systems and Signal Processing* 40 (2) (2013) 667–690.
- [8] Z. Su, L. Ye, Lamb wave propagation-based damage identification for quasi-isotropic CF/EP composite laminates using artificial neural algorithm: Part I – methodology and database development, *Journal of Intelligent Material Systems and Structures* 16 (2005) 97–111.
- [9] Z. Su, L. Ye, Lamb wave propagation-based damage identification for quasi-isotropic CF/EP composite laminates using artificial neural algorithm: Part II – implementation and validation, *Journal of Intelligent Material Systems and Structures* 16 (2005) 113–125.
- [10] K. Worden, Rayleigh and Lamb waves – Basic principles, *Strain* 37 (4) (2001) 167–172.
- [11] I.A. Viktorov, *Rayleigh and Lamb waves – Physical Theory and Applications*, Plenum Press, New York, 1967.
- [12] J.L. Rose, *Ultrasonic Waves in Solid Media*, Cambridge University Press, Cambridge, 1999.
- [13] P.P. Delsanto, T. Whitcombe, H.H. Chaskelis, R.B. Mignogna, Connection machine simulation of ultrasonic wave propagation in materials I: the one-dimensional case, *Wave Motion* 16 (1) (1992) 65–80.
- [14] P.P. Delsanto, R.S. Schechter, H.H. Chaskelis, R.B. Mignogna, R. Kline, Connection machine simulation of ultrasonic wave propagation in materials II: the two-dimensional case, *Wave Motion* 20 (4) (1994) 295–314.
- [15] P.P. Delsanto, R.S. Schechter, R.B. Mignogna, Connection machine simulation of ultrasonic wave propagation in materials III: the three-dimensional case, *Wave Motion* 26 (1997) 329–339.
- [16] (<http://www.vallen.de>).
- [17] B.C. Lee, W.J. Staszewski, Modelling of lamb waves for damage detection in metallic structures: part I. Wave propagation, *Smart Materials and Structures* 12 (2003) 804–814.
- [18] B.C. Lee, W.J. Staszewski, Modelling of Lamb waves for damage detection in metallic structures: part II. Wave interactions with damage, *Smart Materials and Structures* 12 (2003) 815–824.
- [19] G. Dobie, A.B. Spencer, K. Burnham, S.G. Pierce, K. Worden, W. Galbraith, G. Hayward, Simulation of ultrasonic Lamb wave generation, propagation and detection for a reconfigurable air-coupled scanner, *Ultrasonics* 51 (2011) 258–269.
- [20] I. Bartoli, F. Lanza di Scalea, M. Fateh, E. Viola, Modeling guided wave propagation with application to the long-range defect detection in railroad tracks, *NDT&E International* 38 (2005) 325–334.
- [21] P. Packo, T. Bielak, A.B. Spencer, W.J. Staszewski, T. Uhl, K. Worden, Lamb wave propagation modelling and simulation using parallel processing architecture and graphical cards, *Smart Materials and Structures* 21 (2012) 075001.
- [22] I.T. Nabney, *NETLAB Algorithms for Pattern Recognition*, Springer-Verlag, London, 2004.
- [23] (<http://www1.aston.ac.uk/eas/research/groups/ncrg/resources/netlab/>).
- [24] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
- [25] C.M. Bishop, *Neural Network for Pattern Recognition*, Oxford University press, Oxford, 1995.
- [26] D.C. Montgomery, G.C. Runger, *Applied Statistics and Probability for Engineers*, 4th ed. John Wiley and Sons, 2007.
- [27] S.G. Pierce, K. Worden, A. Bezazi, Uncertainty analysis of a neural network used for fatigue lifetime prediction, *Mechanical Systems and Signal Processing* 22 (2008) 1395–1411.