Contents lists available at ScienceDirect

Chinese Journal of Aeronautics

Journal homepage: www.elsevier.com/locate/cja

Impact characterization on thin structures using machine learning approaches

Abstract

In the present manuscript Machine Learning algorithms are trained and compared to identify and to characterise the impact on typical aerospace panels of different geometry. Experimental activities are conducted to build a proper impacts' dataset. Polynomial regression algorithm and artificial neural network are applied and optimised to panels without stringer to test their capability to identify the impacts. Subsequently, the algorithms are applied to panels reinforced with stringers that represent a significant increase of complexity in terms of dynamic features of the system to test: the focus is not only on the impact position's detection but also on the event's severity. After the identification of the best algorithm, the corresponding Machine Learning model is deployed on an ARM processor mini-computer, implementing an impact detection system, able to be installed on board an aerial vehicle, making it a smart aircraft equipped with an Artificial Intelligence decision-making system.

Keywords: Artificial Neural Network; Impact Localisation; Machine Learning; Polynomial Regression; Structural Health Monitoring

1. Introduction

Structural Health Monitoring (SHM) in the aerospace field has reached a certain level of maturity for what concerns the methods and the sensors that can be used for the damage detection in the different structural parts of an aircraft. Ricci et al.[1] provide a review of the main SHM techniques applied in the aviation based on the propagation of guided waves. Numerical, analytical and experimental characterization of damaged structures is discussed. Despite the different geometries involved with variable levels of irregularities, the tomographic method consisting of an unsupervised threshold definition and a multi-parameter processing is the most effective in detecting and locating the impact damage. Significant results are obtained even when thickness increase, stiffeners or both are located close to the monitored area. The Multi-Parameter Approach taken into consideration, indeed, may be generalised to different scenarios where the features mostly influenced by the defect may be specialized according to the different damage occurrence's generation (i.e., impact, fatigue or others). Rocha et al.[2] represent the state of the art on SHM of aerospace composites with a special focus on the most effective sensors, commercially available and developed in laboratory. Working principles, properties, embedding capabilities, their interaction with host materials are discussed. General considerations about

the influence of the aerospace environment on the development of SHM systems are also formulated: one of the conclusions of the manuscript is that piezo-sensors play still today a relevant role in the damage detection of lightweight structures. Other recent works, related to SHM applied to composites and aerospace field, focus on Lamb waves propagation and possible techniques that can be adopted for the damage identification of typical structures[3–8]. The need to implement a wireless sensors' network on operating aircrafts that would provide real-time information about the integrity of primary structures was studied in[9] and the possible ways to feed the SHM sensors exploiting aircraft structural behaviour for energy harvesting purposes were presented by Agnes[10]. Key aspects were analysed[11] for the actual incorporation of SHM systems into composite aircraft structures, like the need for a holistic damage assessment of the aircraft's structural health and an actual upscaling of these systems to realistic in-service structures. Impacts can be detrimental to the aerospace structures and scientists pay significant efforts, on one side, to improve the materials' impact properties[12–16] and, on the other side, to detect impact related damage[17–19]. Machine learning algorithms are nowadays attractive and powerful tools in different fields of the engineering because they allow solutions of problems without a a-priori knowledge of the whole physical frame behind the problems[20–22]. Different papers, recently issued, report effective approaches based on machine learning for detection of damage on thin-walled structures: Salehi et al.[23] present a robust strategy for the damage identification in aircraft structures, exploiting discrete time-delayed data. The authors develop a novel machine learning framework for damage identification based on the integration of a data fusion model, a low-rank matrix completion, pattern recognition, k-nearest neighbour: simulation results prove the efficiency of the method since the proposed machine learning approach could actually detect the presence and location of damage on a typical wing-stabilizer using time-delayed binary data generated from a network of sensors. Ai et al.[24] present an approach able to automatically detect and localize an impact that may occur in flight on a typical aircraft structure: random forest and deep learning, applied on acoustical signals, are employed to train the models for the source location. The results obtained through this novel method are compared to the results obtained using a conventional artificial neural network for the impact localization carried out in previous research. The authors prove that, in this specific case, random forest and deep learning led to better detection performance. Supervised methods would need training data coming from possible damage patterns and, because this is not always feasible for typical aircraft structures, supervised algorithms are difficult to develop for such applications. Shi et al.[25] present a novel unsupervised damage detection method based on machine learning: a statistical model feeding neural networks and a decision-making process based on deep support vector domain description are the core of the method. The results show that the method is able to detect damage in civil structures. Wang et al.[26]propose a kernel extreme learning machine model-based prediction model for the structural response of laminated glass panels under hard body impact. Impact performance of thin concrete slabs can be characterised through the Sobol's Functional Analysis of Variance (FANOVA)[27]: artificial neural network models are combined with probabilistic sampling techniques to calculate the first and second order Sobol's sensitivity indexes.

In the present work a machine learning algorithm is developed able to locate on a thin metal panel the impact position with a remarkable level of accuracy: the main differences with a previous work carried out by authors[28] are that in the present case, (i) the geometry of the tested samples was significantly complicated by the presence of stringers, responsible of waves' scattering through the panel and (ii) the algorithm is also capable to detect the level of severity of the impact (i.e., the energy associated with the event). The excellent agreement between the predictions and the actual impact positions makes the algorithm suitable for SHM purposes and encourage the authors to test it on even more complex structures (curved reinforced panels, thickness variable thin structures and so on).

2. Experimental setup

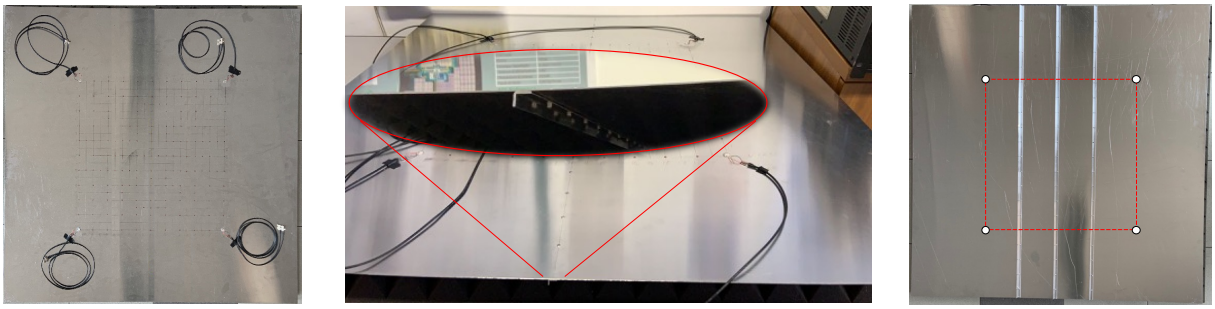
2.1 Characterisation of impacts in the laboratory

In the first activity, low speed impacts were characterised in the laboratory, performing them on three squared specimens:

1. 100×100 cm (Fig. 1 (a));
2. 100×100 cm with "L-shape" single stringer (Fig. 1 (b));
3. 100×100 cm with three "L-shape" stringers (Fig. 1 (c)).

Each panel was made of aluminium alloy with density 2700 kg/m^3, elastic modulus E = 72 GPa, Poisson's ratio ν = 0.33 and thickness = 1.2 mm.

Four piezoelectric ceramic PZT $\mathrm{Pb}[Z_{r_x}\mathrm{T}_{\mathrm{i}_{1-\mathrm{x}}}]\mathrm{O}_3$ sensors[29] (diameter equal to 10 mm) were bonded on the surface, at the four vertices of a square 50×50 cm area (Fig. 1 (a)). The impacts were performed inside the area above by dropping a steel ball from the top of a "drop tower" built up in the AeroSpace Structural Engineering Lab (AS.S.E. Lab – University of Salento)[30]. A steel ball of weight equal to 1.2 g ball weight was dropped from a height of 40 cm at 167 positions identified on a grid of 0.25 mm of step.



(a) 100×100 cm

(b) 100×100 cm with "L-shape" single stringer

(c) 100×100 cm with three "L-shape" stringers

Fig. 1: Specimens.

The waves generated by the impacts were detected by the four PZT sensors, each one directly connected to a separate Picoscope 6402D[31] oscilloscope channel, in order to trace the voltage signals (Fig. 2).

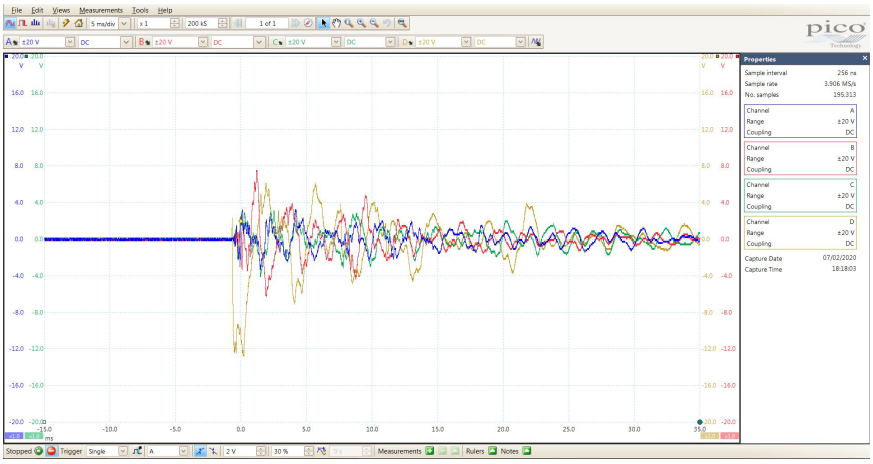


Fig. 2: Signal acquisition in Picoscope software.

The signals were acquired by configuring the Picoscope 6 software[31] as specified below:

- *Timebase control: 5ms/div*
  This control set the time represented by a single division of the horizontal axis (total ten divisions). The acquisition period (50 ms) was set in function of the phenomenon duration (impact and relative plate behaviour), without producing exceedingly large output files.
- *Samples control: 200 kS*
  Number of samples per each channel for the entire acquisition period.
- *Sensor voltage range: +20V (DC)*
  This is the vertical scale with admissible maximum/minimum values.
- *Trigger: Single*
  The Single trigger mode was used to capture all signals waveforms when a selected channel reached a voltage threshold (see next item).
- *Threshold: 2V*
  The voltage at which the trigger operated.
- *Pretrigger: 30%*
  Sets how much time before the trigger event is considered for signal acquisition.

After the acquisition, the signals were sampled and exported as CSV format file using Picoscope software, while MATLAB was used to process the acquired data. In Fig. 3 the lab experimental setup schematic is reported.

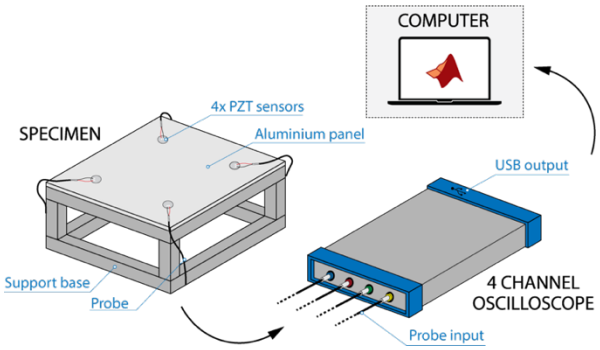


Fig. 3: Experimental setup for impacts characterisation

2.2 *Impact Time of Flight estimation*

An impact produces multiple Lamb wave modes at various frequencies (from zero to infinity) which propagate with different group velocities. At low frequencies (hundreds of kilohertz), only the zero-modes (S0 and A0) can be excited[32,33].

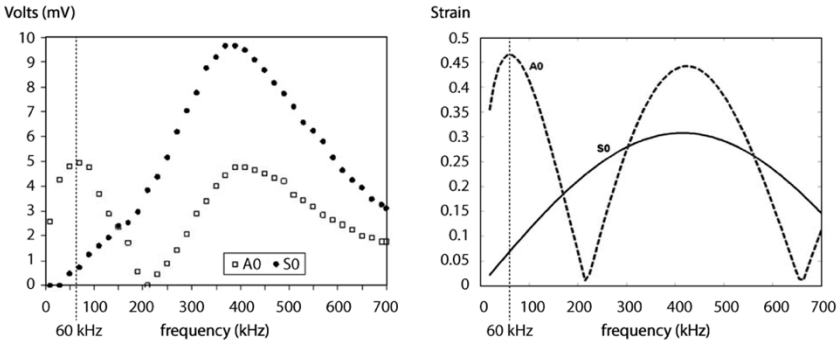


Fig. 4: Lamb waves tuning curves for an aluminium plate of 1.1mm thickness and 7mm square PWAS[32]

Fig. 4 shows the tuning curves for the A0 and S0 modes, experimentally obtained by exciting an aluminium plate of similar thickness to that used in this work[32]. The first maximum of the A0 mode occurs at around 60 kHz, while it is possible to identify in 40 kHz the maximum frequency at which the S0 mode is neglectable. Therefore, in the range 0-40 kHz, the arrival time, also called Time of Flight (ToF), of the entire signal can be considered that of the A0 mode. For this reason, after the acquisition by the oscilloscope, the ToF was evaluated by processing the signals in MATLAB by a Short Time Fourier Transform (STFT) in the range 0-40 kHz, using Hamming window and resolution of 40 Hz (Fig. 5).

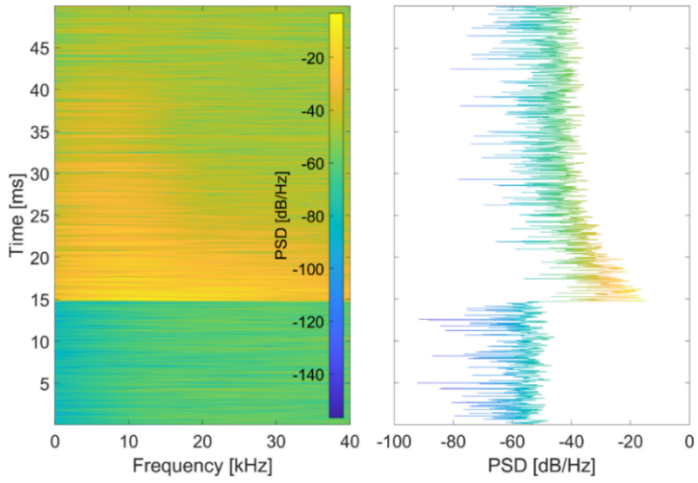


Fig. 5: Signals processed in MATLAB by STFT, range 0 – 40 kHz, Hamming window and 40 Hz resolution.

At the frequency 40 kHz the ToF of the A0 mode was evaluated according to the procedure described by Schindler et al.[34] for each sensor. Because of the absence of an absolute clock signal, it was important to avoid the comparison between ToFs, while the differences t_1, t_2, t_3 between the ToFs at three sensors and the ToF at one reference sensor was a good solution. After the evaluation of the arrival time, a dataset was built, made of 167 samples (impacts points) and, for each sample, the actual coordinates (x,y) and the three ToF differences t_1, t_2, t_3.

2.3 Onboard impact localisation mini-device

In the second experimental activity, an impact localisation mini-device was implemented. This mini-device was made up of two components:

1. an ARM CPU Raspberry Pi minicomputer[35];
2. two BitScope BS05P[36] oscilloscopes connected to the Raspberry Pi.

The best Machine Learning (ML) model identified during the first activity was developed in C++ through MATLAB Coder, compiled as a standalone executable file and uploaded on the Raspberry Pi using MATLAB Support Package. Test impacts were performed on the 100×100 cm plate with three "L-shape" stringers, while the generated waves were detected by the four sensors and processed by the BitScope system. The Raspberry Pi, running an acquisition software developed in Python language and the ML standalone application, was able to predict the impact localisation as expected (Fig. 6).

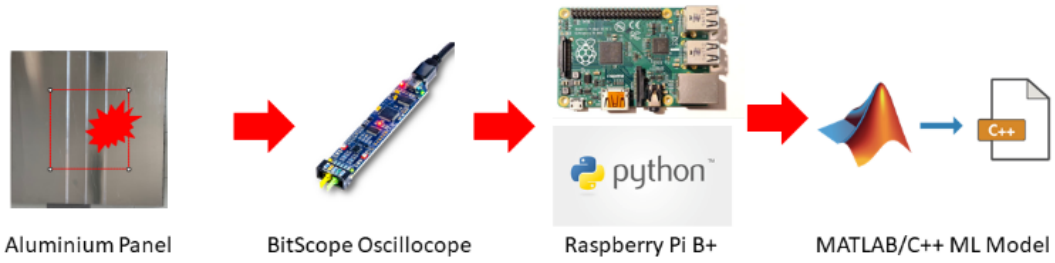


Fig. 6: Experimental setup with onboard impact localisation device.

3. Theory of Machine Learning models

3.1 Polynomial regression

In this work the coordinates (x,y) of an impact in a two-dimensional system are functions of the three ToF differences t_1, t_2, t_3. Applying a polynomial regression algorithm, these functions are represented by polynomials and the construction of the model consists of identifying the degree d and the coefficients θ in order to best fit a set of given data. Per each impact, each coordinate is expressed by a specific function:

$$x = \theta_{x_0} + \sum_{i=1}^{3} \theta_{x_i} t_i + \sum_{j=1}^{3} \sum_{k=1}^{3} \theta_{x_{jk}} t_j t_k + \cdots \quad (1)$$

$$y = \theta_{y_0} + \sum_{i=1}^{3} \theta_{y_i} t_i + \sum_{j=1}^{3} \sum_{k=1}^{3} \theta_{y_{jk}} t_j t_k + \cdots \quad (2)$$

Fixed the polynomial degree d and extending the above equations to L impacts, it is possible to use the following matrix form:

$$\boldsymbol{U} = \boldsymbol{T}\boldsymbol{B} \quad (3)$$

where:

- $\boldsymbol{U}$ is L×2 matrix, in which the columns contain the x and y coordinates for the L impacts respectively;
- $\boldsymbol{T}$ is L×p matrix, in which the columns contain the so-called polynomial features (t_i, power of the t_i up to grade d and relative cross multiplication);
- $\boldsymbol{B}$ is p×2 Design Matrix of weight coefficients θ.

A subset of M impacts was used as *training* data, in order to calculate the design matrix $\boldsymbol{B}$ as specified by Bishop[20], by using MATLAB *fitlm* function, that solves a fitting regression problem with QR decomposition:

$$\boldsymbol{B} = (\boldsymbol{T}^T\boldsymbol{T})^{-1}\boldsymbol{T}^T\boldsymbol{U} \quad (4)$$

A Rank-Deficient error occurred during the calculation of the inverse matrix in (4), due to the correlation between the polynomial features and the consequent ill-conditioning of the $\boldsymbol{T}$ matrix.

The features were normalized applying the *Z-Score feature scaling* on over the samples in this way:

$$t_{i_{NEW}} = \frac{t_i - t_{mean}}{dev} \tag{5}$$

where t_{mean} is the mean value and *dev* is the standard deviation, both calculated by using all samples. Therefore, each new t_i is centred and scaled to have mean value 0 and standard deviation 1.

After the calculation of the $\boldsymbol{B}$ matrix, it was possible to validate the model by estimating the Mean Radial Error (MRE) over the training set of *M* samples and over the test set of remaining *N* samples:

$$RE_i = \sqrt{(\bar{x}_i - x_i)^2 + (\bar{y}_i - y_i)^2} \tag{6}$$

$$MRE_{train} = \frac{1}{M}\sum_{i=1}^{M} RE_i \tag{7}$$

$$MRE_{test} = \frac{1}{N}\sum_{i=1}^{N} RE_i \tag{8}$$

The i^{th} Radial Error RE_i is the Euclidean distance between the actual coordinates (x_i,y_i) and the coordinates $(\bar{x}_i,\bar{y}_i)$ calculated by the algorithm for the i^{th} impact. In order to confirm the choice of the best polynomial degree and evaluate the model in terms of generalisation, a bias-variance diagnosis was performed. *Bias* is the difference between the model average prediction and the expected value, while *variance* is the expectation of squared deviation from its mean. Considering a machine learning model, high bias means that the model is too simple (low complexity), with faster training but also data underfitting. Similarly, high variance means that the model is too complex and too precise in training data fitting (overfitting), causing bad performance and generalisation if applied to test data. Therefore, bias and variance are complementary to each other and the optimum model complexity corresponds to the minimum Mean Square Error with a *bias-variance trade-off*[20].

Considering RE_i as the variable for the evaluation of the bias-variance trade-off, bias is equal to model average prediction (MRE) because of RE_i expected value is equal to 0 for all the samples. It is possible to define:

- Square Error for the i^{th} impact:
$$SE_i = RE_i^2 = (\bar{x}_i - x_i)^2 + (\bar{y}_i - y_i)^2 \tag{9}$$
- Variance:
$$VAR = \frac{1}{L}\sum_{i=1}^{L}[\![(RE]\!]_i - MRE)^2 \tag{10}$$
- Mean Square Error:
$$MSE = \frac{1}{L}\sum_{i=1}^{L} SE_i = BIAS^2 + VAR + IE \tag{11}$$

where *IE* is the irreducible error due to noise.

3.2 Artificial neural network

The other machine learning model applied to the dataset described in the previous section was the Artificial Neural Network (ANN). An ANN is made of connected nodes called artificial neurons (Fig. 7), each of which receives input signals, processes them and sends the output to the other neurons connected to it. The output *a* of a generic neuron is computed based on two consecutive operations:

1. the computation of an intermediate output z using the input vector t, the vector of weights w and the vector of biases b;
2. the activation on z to give out the final output a of the neuron.

$$z = w^T t + b \tag{12}$$

$$a = g(z) \tag{13}$$

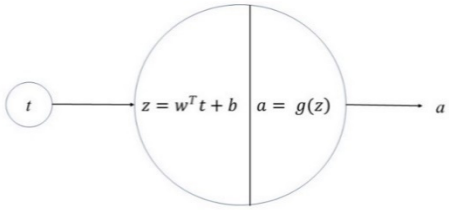


Fig. 7: Neuron representation.

In ANN the neurons are aggregated into layers. Signals travel from the first layer (the input layer, made up of the input variables) to the last layer (the output layer) even more than once, traversing one or more hidden layers, in which the variables are not visible and that contain the neurons (Fig. 8).

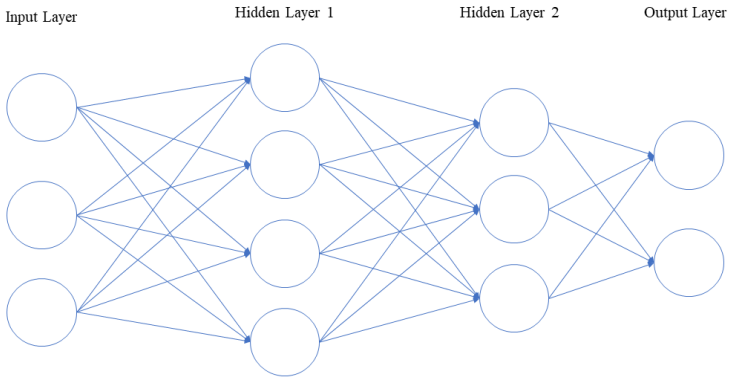


Fig. 8: Neural Network Example with 2 hidden layers.

A simple ANN is the Shallow Neural Network (SNN), a feed-forward network consisting of only one hidden layer (Fig. 9).

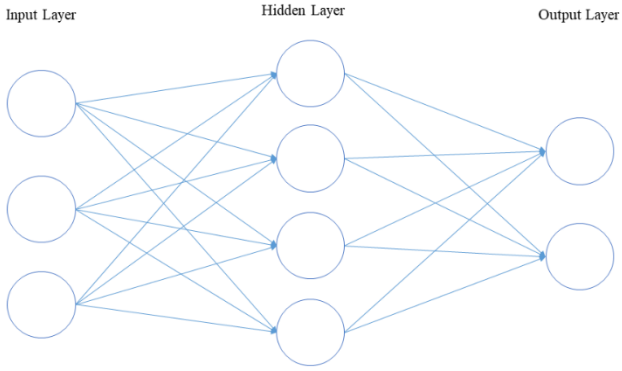


Fig. 9: Shallow Neural Network example.

Considering a SNN in which the input is represented by the three ToF differences t_1, t_2, t_3, a generic neuron identified by l subscript performs the following equations:

$$z_l = w_l{}^T t + b_l \qquad (14)$$
$$a_l = g(z_l)$$

where:

- t is the input vector containing the three ToF differences t_1, t_2, t_3;
- w_l is the vector of three weights associated to the neuron l;
- b_l is the bias associated to the neuron l;

- z_l is the intermediate output associated with the neuron l;
- g is the activation function for the layer;
- a_l is the final output associated with the neuron l.

Using a matrix notation, the above equations can be written as:

$$Z^{[h]} = W^{[h]^{\mathrm{T}}} t + b^{[h]} \qquad A^{[h]} = g^{[h]}\left(Z^{[h]}\right) \tag{15}$$

where:

- $Z^{[h]}$ is the vector containing the intermediate output in the hidden layer;
- $W^{[h]}$ is the matrix containing the weights of the neurons in the hidden layer;
- $b^{[h]}$ is the vector containing the biases of the neurons in the hidden layer;
- $g^{[h]}$ is the activation function for the hidden layer;
- $A^{[h]}$ is output of the hidden layer that becomes the input for the output layer.

A similar matrix notation can be used for the calculation of the final output, given by the *forward-propagation equations*:

$$Z^{[o]} = W^{[o]^{\mathrm{T}}} A^{[h]} + b^{[o]} \qquad O = g^{[o]}\left(Z^{[o]}\right) \tag{16}$$

in which:

- $Z^{[o]}$ is the vector containing the intermediate output in the output layer;
- $W^{[o]}$ is the matrix containing the weights of the neurons in the output layer;
- $b^{[o]}$ is the vector containing the biases of the neurons in the output layer;
- $g^{[o]}$ is the activation function for the output layer;
- O is the output containing the coordinates $(\bar{x},\bar{y})$ calculated by the neural network.

A feed-forward shallow neural network can be used for regression problems (data fitting) and for pattern recognition, using the same architecture shown in Fig. 9 with the only difference represented by the activation function. For example, considering a regression problem, the activation function g must be a continuous function, like a linear function, a *tanh* function or a sigmoid function in both layers, while, for pattern recognition, a discrete function like *softmax* must be used in the output layer. The sigmoid function is mathematically defined as:

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{17}$$

The *softmax* function is a function that turns a vector of K real values into a vector of K real values that sum to 1, so, while the input values can be positive, negative, zero, or greater than one, the output are values between 0 and 1 and can be interpreted as probabilities. If an input is small or negative, the *softmax* turns it into a small probability, while if an input is large, the corresponding output is a large probability. The *softmax* function is mathematically defined as:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{18}$$

where all the z_i are the elements of the input vector z and can take any real value. The term on the bottom of the formula is the normalization term which ensures that all the output values of the function will sum to 1. K is the number of classes in the multi-class classifier. The *softmax* function and the sigmoid function are similar. The *softmax* operates on a vector while the sigmoid takes a scalar and can be considered a special case of the *softmax* function for a classifier with only two input classes.

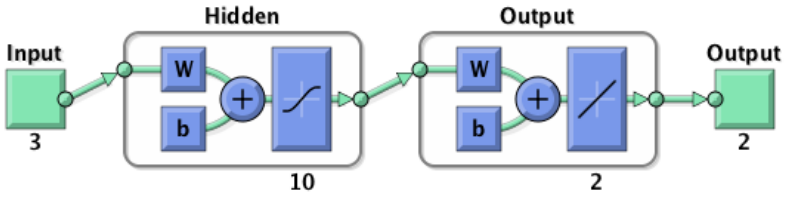


Fig. 10: MATLAB representation of a Shallow Neural Network used for data fitting, with two layers, a sigmoid transfer function in the hidden layer and a linear transfer function in the output layer[37].

Fig. 10 and Fig. 11 show a MATLAB representation of a Shallow Neural Network used for data fitting (linear transfer function in the output layer) and for pattern recognition (*softmax* transfer function in the output layer) respectively.

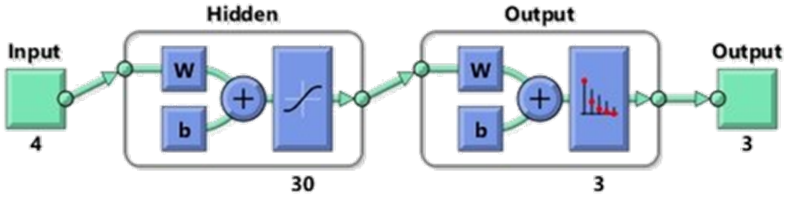


Fig. 11: MATLAB representation of a Shallow Neural Network used for pattern recognition, with two layers, a sigmoid transfer function in the hidden layer and a softmax transfer function in the output layer[37].

The shallow neural network is a supervised learning model in which, after the initialisation of weights and biases (matrixes $W^{[h]}$ and $W^{[o]}$ and vectors $b^{[h]}$ and $b^{[h]}$), for example according to Xavier[38], a training phase must be performed in order to set the value of weights and biases on the basis of a training subset data with known output. This phase is an iterative procedure implemented by means of several possible algorithms[38–40].

Each training algorithm contains:

1. a forward step that performs the forward-propagation equations (15) and (16);
2. a backward step that updates the weights and thresholds values to close in on the known output of training data.

Three learning algorithms were compared.

- Levenberg-Marquardt;
- Bayesian Regularization;
- Scaled Conjugate Gradient.

The Levenberg-Marquardt is a second order training algorithm used for non-linear optimization problems with faster convergence[39]. It considers a loss function F represented by a sum of squared errors:

$$F = \sum e_i^2 \tag{19}$$

Here e_i is the error (distance between real and predicted data) for the impact i and the sum is computed on all training samples. At each iteration k:

1. the Jacobian matrix J of the loss function F is calculated deriving each error e_i with respect to each weight and bias of the neural network;
2. the algorithm adjusts the weights as shown in the expression below:

$$w_{k+1} = w_k - (J_k^T J_k + \alpha_k I)^{-1}(J_k^T e_k) \tag{20}$$

where w_k is the vector of weights and biases at iteration k, α_k is the learning rate and I is the identity matrix. The Bayesian Regularization algorithm provides better solution and generalization, at the cost of taking longer. The update of weights and biases and the minimization of the loss function F are similar to that of the Levenberg-Marquardt method, but the problem is converted to a statistical problem, involving a probability distribution of weights[39].

The Scaled Conjugate Gradient algorithm was developed by Moller[40], that simplified the other conjugate gradient methods by removing the line search at each iteration. Therefore, this method avoids the time-consuming line search and just searches along conjugate directions which lead to faster convergence.

4. Results and discussions

4.1 Specimen 100×100 cm

Considering the 100×100 cm panel, in a first analysis the polynomial regression model was applied to the dataset. The B calculation and the corresponding MRE estimating were performed by the mean in a K-Fold cross validation procedure, considering 5 different combinations of training/test sets with an 80/20 ratio. This procedure was applied to polynomial models with different degrees: Fig. 12 shows MRE trends for (i) entire dataset (yellow curve), (ii) training set of 80% (blue curve) and (iii) test set of 20% (red curve) of it.

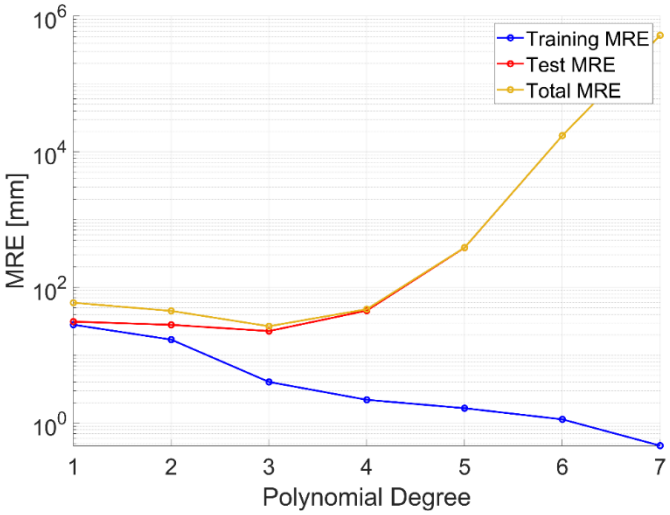


Fig. 12: Radial Error vs Complexity for 100×100 cm panel.

The best model, in terms of generalization, was chosen considering both minimum total MRE and minimum gap between training and test MREs. This condition occurs with degree equal to 3. So, the degree equal to 3 was chosen for the analysis related to best training set size, as shown in Fig. 13.

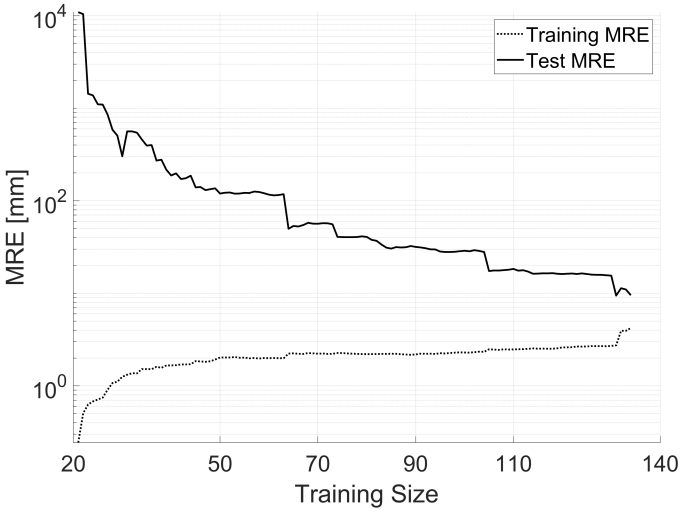


Fig. 13: Radial Error vs Training size for 100×100 cm panel.

By keeping the same test set size of the previous analysis (20% of the entire dataset), it's clear that it's necessary to train the model with the maximum number of samples, that is 134 (80% of the entire dataset). After the evaluation of the training size, a bias-variance diagnosis was performed in order to confirm the best polynomial degree of the model (Fig. 14).

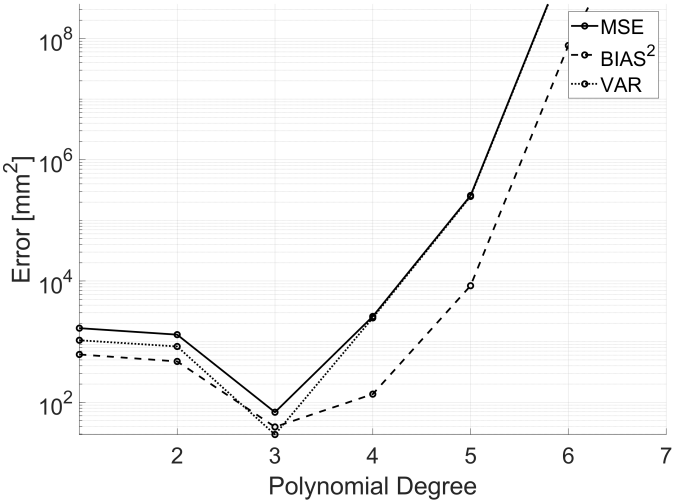


Fig. 14: Bias-Variance trade-off vs Complexity.

As shown in Fig. 14, the degree equal to 3 is the best in terms of error and generalisation, because it corresponds to the minimum MSE, BIAS^2 and VAR; lower and higher degrees lead to underfitting or overfitting, as explained by Bishop[20]. After the bias-variance diagnosis and then of the best choice of polynomial degree and training size, a 50 tests campaign was performed, with the calculation of Mean Radial Error (MRE) on an average of 50 cases with polynomial degree equal to 3 and the following random division of data:

- training set: 80%;
- test set: 20%.

Different test cases gave very different results, with MRE values from 5.48 mm (the best one) to 27.45 mm (the worst one). It is clear that the ToFs (used as the basis for the machine learning features) are very different from each other and the algorithm "struggles" to find correspondences between them and then to “trace” a curve that best fits such different data. Fig. 15 shows a single test case.

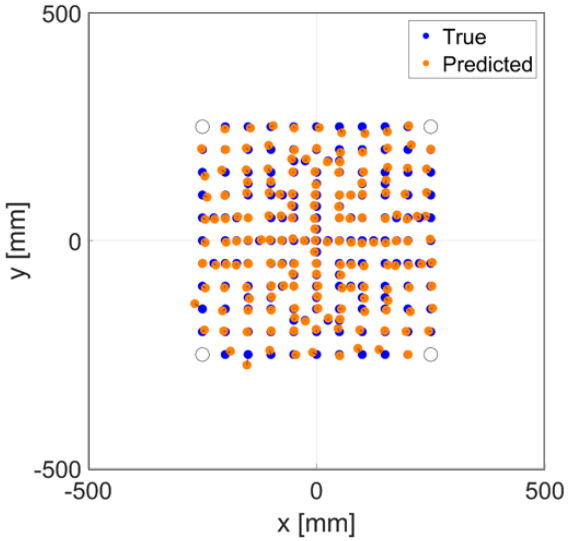


Fig. 15: Regression model test case (blue points for actual impact positions, orange points for predicted ones).

Still considering the 100×100 cm panel, in a second analysis the performances of the shallow neural network were evaluated over the entire dataset, with the following parameters:

- no features scaling;
- two layers with:
  - a sigmoid transfer function in the hidden layer;

  - a linear transfer function in the output layer;
- three learning algorithms:
  - Levenberg-Marquardt;
  - Bayesian Regularization;
  - Scaled Conjugate Gradient;
- maximum number of epochs (iterations): 1000;
- Mean Radial Error (*MRE*) calculated on an average of 50 test cases with the following random division of data:
  - training set: 70%;
  - test set: 30%.

Each network configuration was trained by MATLAB *train* function[37], increasing the complexity of the model in terms of number of neurons in the hidden layer: 10, 20, 30, 40, 50.

Table 1: Shallow neural network 100×100 cm panel best results

Number of neurons	Training method	MRE [mm]
30	Levenberg-Marquardt	6.74
30	Bayesian Regularization	3.08
20	Scaled Conjugate Gradient	26.06

The best result was obtained by training with Bayesian Regularization algorithm and 30 neurons, with a MRE equal to **3.08 mm**, as shown in Table 1, while Fig. 16 shows a test case.

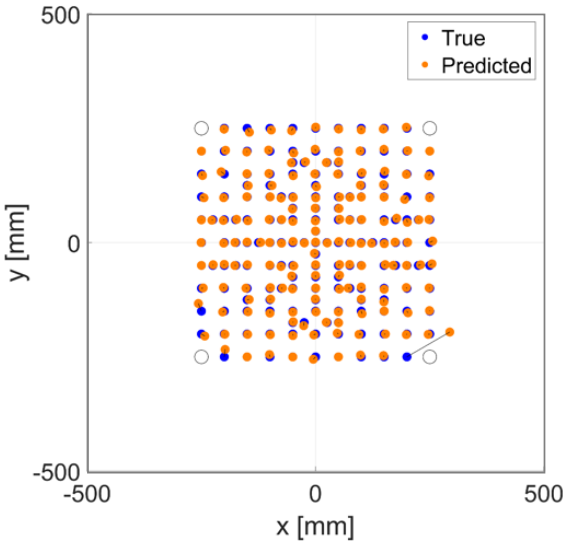


Fig. 16: Shallow neural network model test case for a 100×100 panel with Bayesian Regularization training (blue points for actual impact positions, orange points for predicted ones).

4.2 Specimen 100×100 cm with "L-shape" single stringer

Subsequently, the same neural network and the same comparisons were applied to the experimental data obtained from the impacts on a 100×100 panel with a 1 mm thickness "L-shape" aluminium stringer fixed in the centre.

Table 2: Shallow neural network 100×100 cm panel with "L-shape" single stringer best results

Number of neurons	Training method	MRE [mm]
30	Levenberg-Marquardt	9.86
30	Bayesian Regularization	**4.24**
20	Scaled Conjugate Gradient	28.10

As in the case of the 100×100 panel without stringer, the best result was obtained training with Bayesian Regularization algorithm and 30 neurons, with an MRE equal to **4.24 mm**, as shown in Table 2, while Fig. 17 shows a test case.

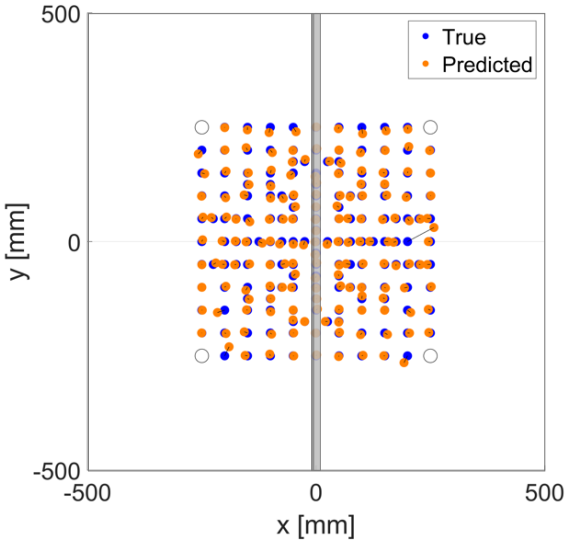


Fig. 17: Shallow neural network model test case for a 100×100 panel with a 1 mm thickness "L-shape" stringer and Bayesian Regularization training (blue points for actual impact positions, orange points for predicted ones).

Therefore, the Bayesian Regularization training confirms its effectiveness and precision[39], at the cost of more complex algorithm and longer execution time. In Table 3 a deep comparison between the two scenarios (100×100 cm panel with and without stringer), with the same neural network configuration and the same analysis, is shown.

Table 3: Error comparison between panels

Without Stringer			With Stringer		
Mean Error	Min Error	Max Error	Mean Error	Min Error	Max Error
3.09 mm	1.93 mm	6.18 mm	4.24 mm	3.16 mm	6.44 mm

The presence of the reinforcement worsens the error, even if the mean one (4.24 mm) falls between the minimum (3.09 mm) and the maximum error (6.18 mm) of the case without reinforcement. The reason for the deterioration in performance lies in the fact that the features used by the machine learning algorithm are not consistent in the two scenarios:

- without stringer, all the ToFs are obtained from direct impact-sensor waves;
- with stringer,
    - 2 out of 4 ToFs are obtained from direct impact-sensor waves;
    - 2 out of 4 ToFs are obtained from impact-reinforcement-sensor «transmitted» waves.

In order to quantify the difference described above, an impact located in the "left" side and a sensor bonded in the "right" size of the panel of the panel were considered. The choice of the impact-sensor pair was made evaluating all the dataset, so as to identify a significant feature (ToF difference between the sensor and a reference one).

The M2 impact location and the 4th sensor were chosen (impact number 105/167), to have a distance (that is the wave path) of 49 cm (Fig. 18). The ToF without stringer was equal to 14.650 milliseconds, while the ToF with stringer was equal to 14.648 milliseconds; in absence of a reference clock signal however only their difference is significant, that is equal to **2.07 microseconds**. Therefore, in the presence of reinforcement, the arrival time is shorter and the wave (orange plot) is anticipated in time with respect to the propagation without reinforcement (blue plot), as shown in MATLAB representation (Fig. 19).

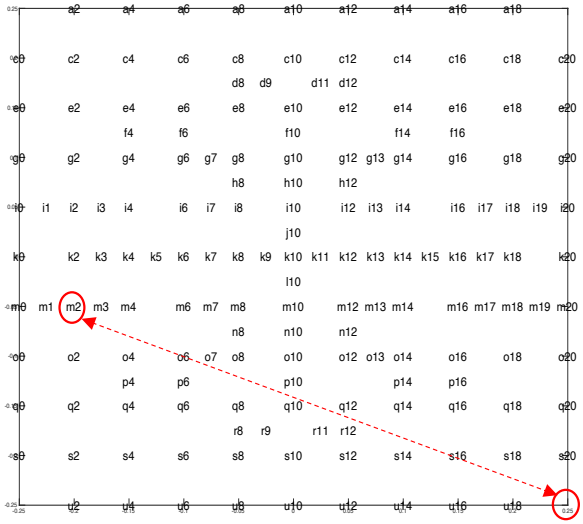


Fig. 18: Impact-sensor pair.

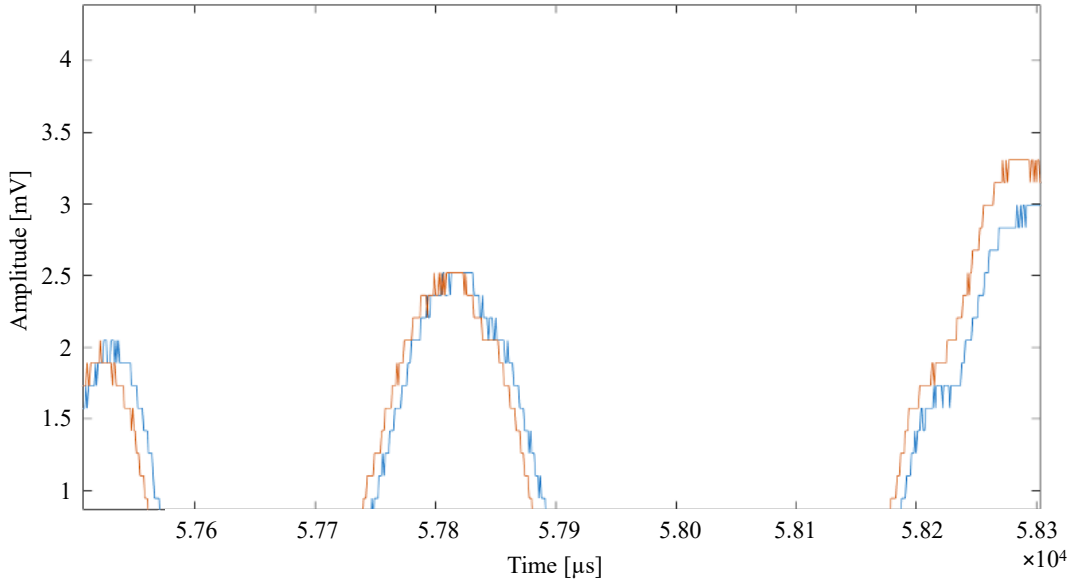


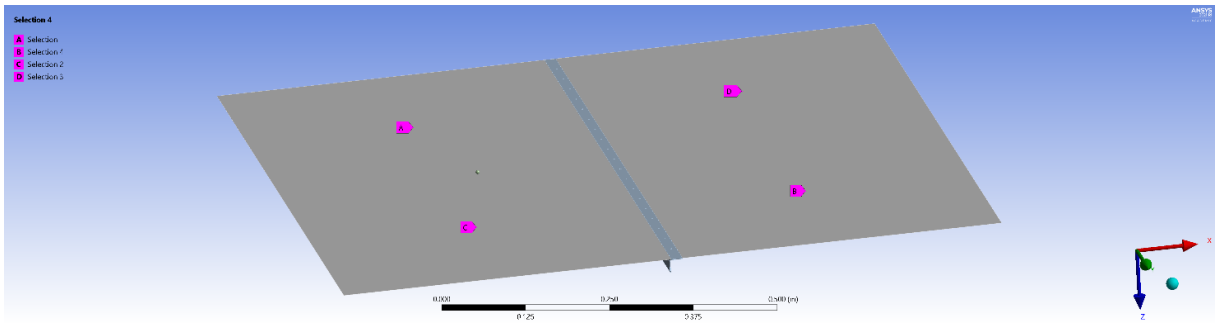
Fig. 19: Zoom of differences between signals traveling on simple plate (blue) and on reinforced plate (orange).

The behaviour described below was confirmed by a simulation with dynamic characterization made in ANSYS (Fig. 20 and Fig. 21), where three scenarios were considered:

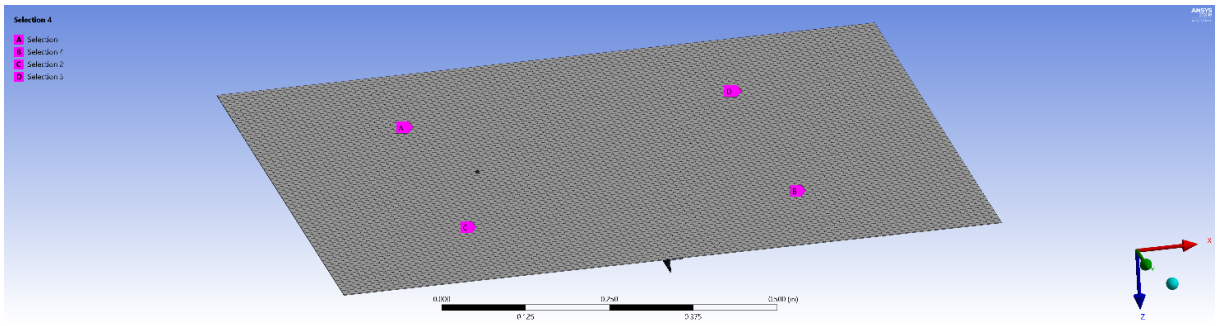
1. absence of reinforcement;
2. "glued" reinforcement;
3. reinforcement fixed with rivets.

Fig. 22 shows that the red and orange waveforms (propagation with reinforcement) are anticipated if compared to the blue one (propagation without reinforcement) and then the peaks "come first", with consequent shorter arrival time.

By concentrating the analysis on A0 and S0 modes, the impact on the reinforcement generates reflected and transmitted modes, that travel at the same speed as the direct modes in the propagation inside the plate[41]. The explanation of the different arrival time is that the wave (and therefore the A0 mode on which the ToF is measured) travels faster only within the reinforced area, where the thickness has doubled.

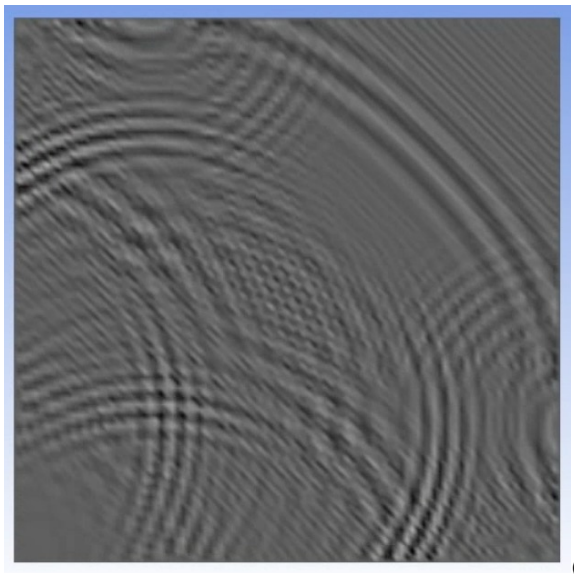


(a)

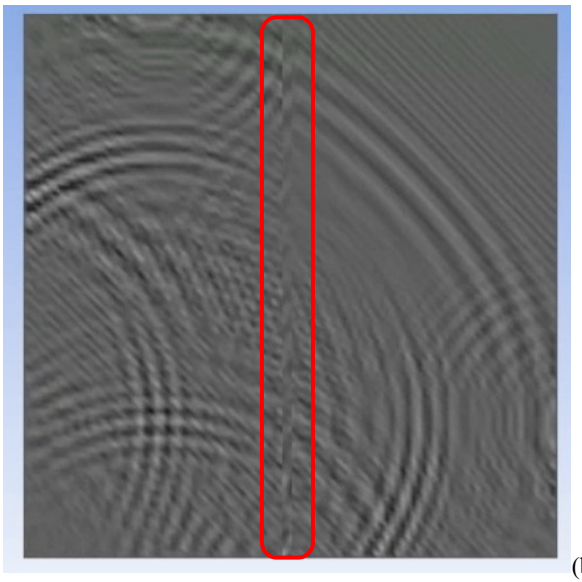


(b)

Fig. 20: Finite elements model of reinforced plate.



(a)



(b)

Fig. 21: Finite elements simulations to highlight the different wave propagations without (a) or with (b) the stringer.

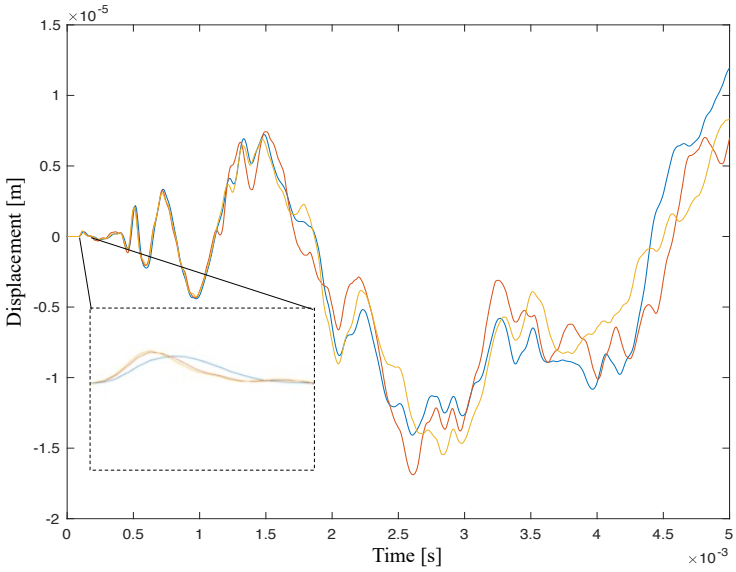


Fig. 22: Differences between numerical signals traveling from impact location M2 to one sensor directly (blue plot) and through the stiffener (orange and red plots).

Further confirmation was provided in MATLAB by the computation of dispersion curves, in which, considering the propagation of A0 mode at 40 kHz (frequency chosen for the ToF evaluation), the following speeds were calculated:

- in the plate zone without stringer (thickness of 1.2 mm): 1135 m/s;
- in the plate zone with stringer (thickness of 2.2 mm): 1590 m/s.

Considering the M2 impact, its distance from 4th sensor is equal to 49 cm, of which 48 cm in the plate zone without stringer and 1 cm in the plate zone with stringer. According to the speed provided by the dispersion curves, the following propagation times were obtained:

- plate without stringer: 49 cm in 431.72 microseconds;
- plate with stringer: 49 cm in 429.20 microseconds, of which:
  - in the plate zone without stringer: 48 cm in 422.91 microseconds;
  - in the plate zone with stringer: 1 cm in 6.29 microseconds.

Summarizing, the total arrival time without stringer was equal to 431.72 microseconds, while the total arrival time with stringer was equal to 429.20 microseconds, with **2.52 microseconds** of difference. This result is very close to the ToF difference calculated from the experimental measurement (**2.07 microseconds**): it confirms the explanation above.

4.3 Specimen 100×100 cm with three "L-shape" stringers

4.3.1 Localisation

Subsequently, the shallow neural network was applied to the experimental data obtained from the impacts on a 100×100 panel with three "L-shape" aluminium stringers (1 mm thickness) fixed at equally spaced positions.

Table 4: Shallow neural network error for a 100×100 panel with three "L-shape" stringers with Bayesian Regularization training

Number of Neurons	Mean Error	Min Error	Max Error
30	8.88 mm	6.29 mm	14.81 mm
40	8.80 mm	5.75 mm	13.94 mm

In this scenario, with Levenberg-Marquardt and Scaled Conjugate Gradient training algorithms the MRE largely exceeded 15 mm for any number of neurons, while, as in the previous cases, the best result was obtained training with Bayesian Regularization algorithm but with a number of neurons equal to 40, as detailed in Table 4. Fig. 23 shows a test case.

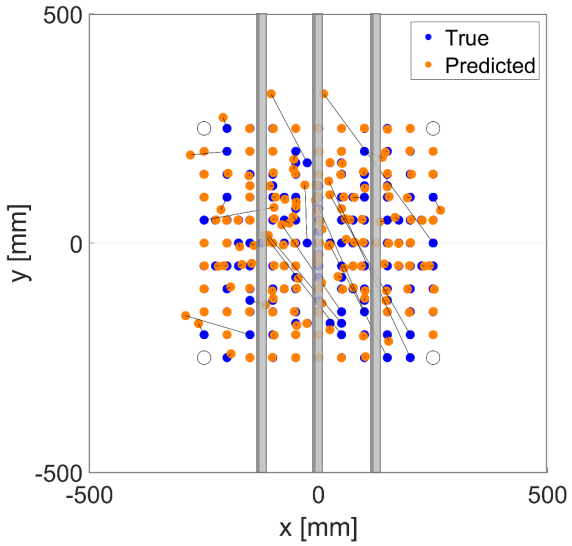


Fig. 23: Shallow neural network model test case for a 100×100 panel with three 1 mm thickness "L-shape" stringer and Bayesian Regularization training (blue points for actual impact positions, orange points for predicted ones).

Therefore, in presence of three stringers, the phenomenon explained before is "multiplied", causing about twice the error. In order to improve the prediction, 40 neurons were kept in the hidden layer, but another hidden layer was added to the shallow neural network, obtaining a custom neural network (Fig. 24). This more complex network was applied to the dataset, with the following parameters:

- Z-Score features scaling;
- three layers with:
  - a sigmoid transfer function in the first hidden layer and in the second one;
  - a linear transfer function in the output layer;
- Bayesian Regularization learning algorithms;
- maximum number of epochs (iterations): 1000;
- Mean Radial Error (MRE) calculated on an average of 50 test cases with the following random division of data:
  - training set: 70%;
  - test set: 30%.

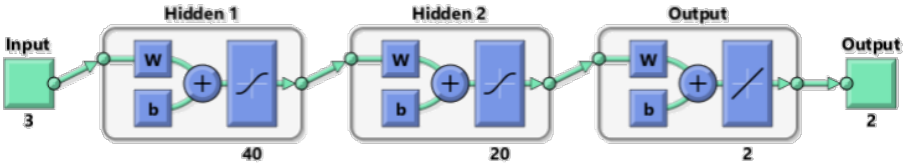


Fig. 24: MATLAB representation of a custom neural network used for data fitting, with three layers, a sigmoid transfer function in the hidden layers and a linear transfer function in the output layer.

Each network configuration was trained by MATLAB *train* function[37], increasing the complexity of the model in terms of number of neurons in the second hidden layer: 10, 20, 30, 40. The results are listed in Table 5, while Fig. 25 shows a test case.

Table 5: Custom neural network performances

2nd layer neurons	Mean error	Min error	Max error
10	7.43 mm	5.06 mm	11.54 mm
20	**6.88 mm**	5.18 mm	8.74 mm
30	7.08 mm	5.23 mm	9.21 mm
40	7.34 mm	4.73 mm	11.33 mm

The best result was obtained with 20 neurons in the second layer, with a MRE equal to **6.88 mm**. Therefore, the increasing in network complexity improved the MRE of about 2 mm.

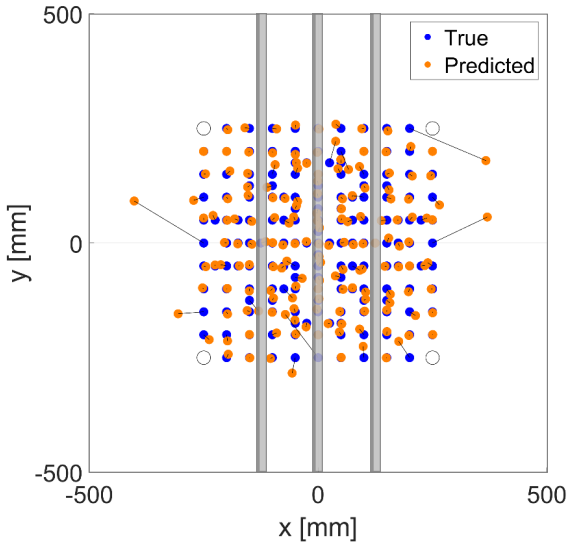


Fig. 25: Custom neural network model test case for a 100×100 panel with three 1 mm thickness "L-shape" stringer and Bayesian Regularization training (blue points for actual impact positions, orange points for predicted ones).

4.3.2 Energy classification

The following analysis was to apply machine learning algorithm in order to predict the energy of an impact. In the third scenario (100×100 panel with three "L-shape" aluminium stringers), according to the grid described in Section 2, 167 impacts were generated at three different heights (10, 20, 40 cm), corresponding to three different increasing energies, with each equal to double the previous one.

Processing the waveforms acquired by the four sensors, for each impact and for each height, the energy of the waves (as power integration) was calculated. In this way, a new database was built and four features became the inputs for the model.

In a first step all previous regression algorithms were applied to this database, but the implemented models were not able to predict the correct energy levels. It is noteworthy that the presented problem (with only three energy levels) can not be solved by a regression approach, but it can be better represented as a pattern recognition one, in which classify inputs according to target classes. Therefore, a feed forward pattern recognition neural network was chosen, with Bayesian Regularization training algorithm, that had provided the best precision in the previous predictions.

The performances of the neural network were evaluated over the entire dataset, with the following parameters

- Z-Score features scaling;
- two layers with:
  - a sigmoid transfer function in the hidden layer;
  - a soft max transfer function in the output layer;
- maximum number of epochs (iterations): 1000;
- Classification Error (*CE*) calculated on an average of 50 test cases with the following random division of data:
  - training set: 70%;
  - test set: 30%.

Each network configuration was trained by MATLAB *train* function[37], increasing the complexity of the model in terms of number of neurons in the hidden layer: 10, 20, 30, 40, 50.

The best result was obtained by training with Bayesian Regularization algorithm and 30 neurons, with a CE equal to **1.2%**. Fig. 26 shows the confusion matrix related to a test case.

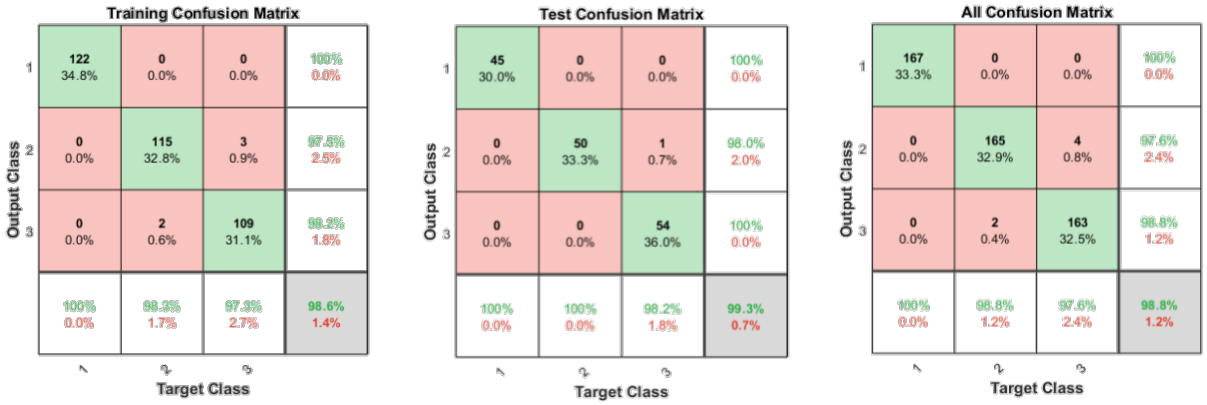
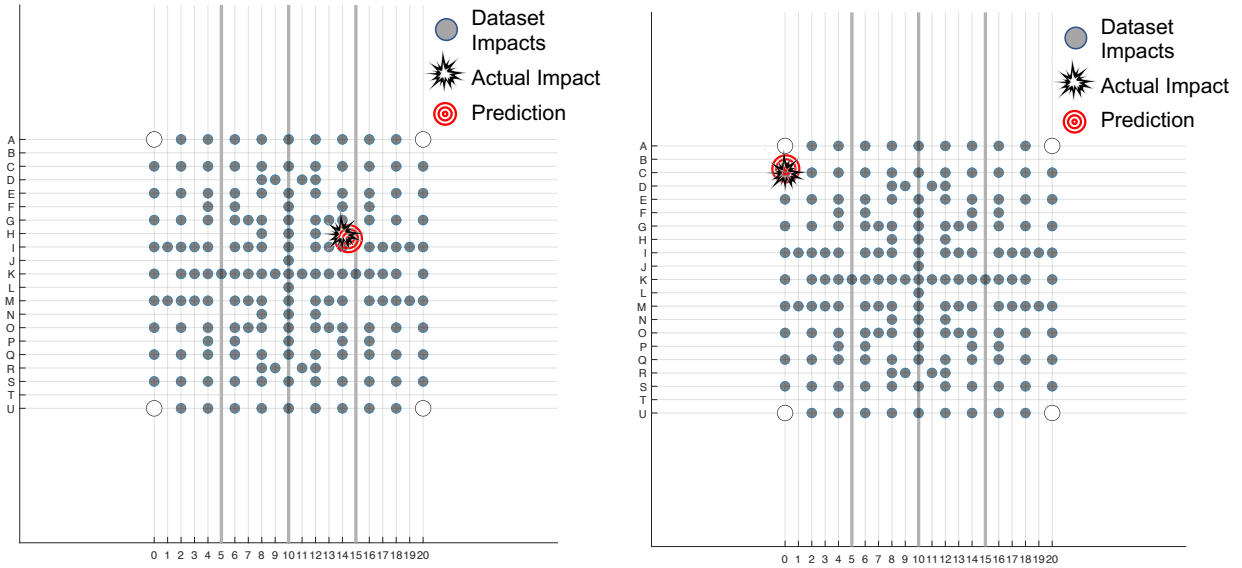


Fig. 26: Confusion matrix of a test case for a feed forward pattern recognition neural network with Bayesian Regularization training and 30 neurons in the hidden layer.

4.4 ML model implementation on impact localisation mini-device

The custom neural network (Fig. 24) identified as the best ML model for the 100×100 panel with three "L-shape" stringers was deployed on the Raspberry Pi based system described above.

An experimental campaign was carried out in order to verify the functionality of the impact localisation mini-device.



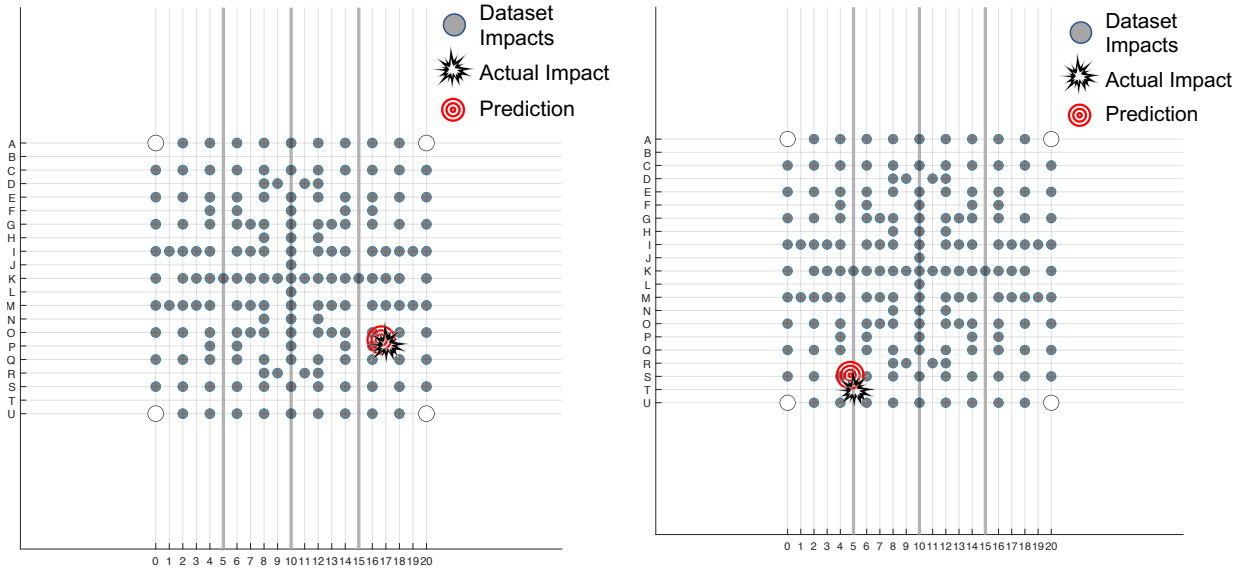


Fig. 27: Custom neural network model test cases for a 100×100 panel with three 1 mm thickness "L-shape" stringers, deployed on onboard impact localisation mini-device (blue points for actual impact positions, orange points for predicted ones).

The presented results in Fig. 27 highlight the reason that make the approach interesting: the ML model provides an excellent output by using Raspberry Pi system, in terms of impact location prediction. Another fundamental aspect concerns weight and power absorption of the mini-detection system. In fact, the Raspberry Pi is connected to two mini BitScope oscilloscopes, with a total weight of 120 g, that is negligible compared to 2.5 kg, i.e., the weight of the primary detection system (laptop and Picoscope oscilloscope). The same comparison affects the current consumption: the Raspberry Pi and the two BitScope oscilloscopes are USB devices (5 V required) with a total max current absorption of 1 A per each device. The total max power absorption is equal to 15 W, that is negligible compared to 100 W max required by of the primary detection system.

In conclusion, in the authors opinion, the elegance, simplicity and low cost of the ML model make it particularly interesting for smart aircraft applications.

4. Conclusions

This work focused on the determining the location of low-speed impacts. The present study is **independent of the material**, just needs four sensors and applies two type of Machine Learning algorithms on three different aluminium plates: 100×100 cm, 100×100 cm with "L-shape" single stringer and 100×100 cm with three "L-shape" stringers. The main conclusions from this study can be summarized as follows.

(1) A dataset of 167 impacts was created and polynomial regression algorithm and shallow neural network were applied to the simple **100×100 cm plate**. The best polynomial degree, in terms of error and generalization, was found to be equal to 3, while the best neural network training algorithm was found to be the Bayesian Regularization.
   a. The best **polynomial regression MRE** was equal to **5.48 mm**.
   b. Applying the **shallow neural network**, the best result was obtained with 30 neurons, with a **MRE** equal to **3.08 mm**.

(2) Only the **shallow neural network** was applied to a **100×100 cm plate with "L-shape" single stringer**, obtaining the best result with Bayesian Regularization algorithm and 30 neurons, with an **MRE** equal to **4.24 mm**.
   a. The presence of the reinforcement worsens the error. The **transmitted waves travel faster** only within the reinforced area, where the thickness has doubled.
   b. The behaviour described below was confirmed by a simulation with dynamic characterization made in **ANSYS** and by **experimental** measurement.

(3) The shallow neural network was applied to the experimental data obtained from the impacts on a **100×100 panel with three "L-shape" aluminium stringers** fixed at equally spaced positions.

a. In presence of three stringers, the phenomenon explained before is “multiplied”, causing about twice the error.
b. In order to improve the prediction, 40 neurons were kept in the hidden layer, but another hidden layer was added to the shallow neural network, obtaining a **custom neural network** that provided a prediction with **MRE** equal to **6.88 mm**.

(4) Eventually, 167 impacts were generated on the **100×100 panel with three “L-shape” aluminium stringers** at three different heights, 10, 20, 40 cm, corresponding to **three different increasing energies**. A feed forward pattern recognition neural network was applied to dataset built above in order to predict the **impact energy** in terms of Classification Error (CE) calculated on an average of 50 test cases, obtaining the best CE equal to **1.2%**.

The results can be considered excellent, in terms of mean radial error, that falls in an acceptable range if compared to the relative grid step in each scenario, and in terms of energy classification error, under 2%. These results demonstrate the validity of the Machine Learning application to the impact characterization: its simplicity and computing efficiency make this approach fully applicable to the processing of acoustic signals in SHM systems.

This work presents the development of an innovative mini-equipment able to detect impacts, process data and predict via ML learning software running on a Raspberry Pi micro-computer. This mini-system can be considered very efficient because of its performance in terms of precision and for its little size. It can be installed on board an aerial vehicle, making it a smart aircraft equipped with an Artificial Intelligence decision-making system.

References

1. Rocha Helena, Semprimoschnig Christopher, Nunes João P. Sensors for process and structural health monitoring of aerospace composites: A review. *Eng Struct* 2021;237:112231. Doi: 10.1016/J.ENGSTRUCT.2021.112231.
2. WANG Xiaopeng, CAI Jian, ZHOU Zhiquan. A Lamb wave signal reconstruction method for high-resolution damage imaging. *Chinese Journal of Aeronautics* 2019;32(5):1087–99. Doi: 10.1016/j.cja.2019.03.001.
3. TOWSYFYAN Hossein, BIGURI Ander, BOARDMAN Richard, BLUMENSATH Thomas. Successes and challenges in non-destructive testing of aircraft composite structures. *Chinese Journal of Aeronautics* 2020;33(3):771–91. Doi: 10.1016/j.cja.2019.09.017.
4. WANG Yiwei, GOGU Christian, BINAUD Nicolas, BES Christian, FU Jian. A model-based prognostics method for fatigue crack growth in fuselage panels. *Chinese Journal of Aeronautics* 2019;32(2):396–408. Doi: 10.1016/j.cja.2018.11.010.
5. REN Yuanqiang, XU Qiuhui, YUAN Shenfang. Improving accuracy of damage quantification based on two-level consistency control of PZT layers. *Chinese Journal of Aeronautics* 2023;36(3):241–53. Doi: 10.1016/j.cja.2022.09.021.
6. Hassani Sahar, Mousavi Mohsen, Gandomi Amir H. Structural Health Monitoring in Composite Structures: A Comprehensive Review. *Sensors 2022, Vol 22, Page 153* 2021;22(1):153. Doi: 10.3390/S22010153.
7. Nicassio F., Carrino S., Scarselli G. Elastic waves interference for the analysis of disbonds in single lap joints. *Mech Syst Signal Process* 2019;128:340–51. Doi: 10.1016/J.YMSSP.2019.04.011.
8. Aqueeb Ahsan, Yang Mijia, Braaten Benjamin, Burczek Ellie, Roy Sayan. A Real-Time and Wireless Structural Health Monitoring Scheme for Aerospace Structures Using Fibre Bragg Grating Principle. *Photonics, Plasmonics and Information Optics* 2021:241–66. Doi: 10.1201/9781003047193-9.
9. Elahi Hassan. The investigation on structural health monitoring of aerospace structures via piezoelectric aeroelastic energy harvesting. *Microsystem Technologies 2020 27:7* 2020;27(7):2605–13. Doi: 10.1007/S00542-020-05017-Y.
10. Broer Agnes A.R., Benedictus Rinze, Zarouchas Dimitrios. The Need for Multi-Sensor Data Fusion in Structural Health Monitoring of Composite Aircraft Structures. *Aerospace 2022, Vol 9, Page 183* 2022;9(4):183. Doi: 10.3390/AEROSPACE9040183.
11. Zhang Huawen, He Rui, Hou Bing, Li Yulong, Cui Hao, Yang Wei. Artificial hail ice impact damage of laminated composite T-joint with stitching reinforcement. *Compos Struct* 2021;278:114714. Doi: 10.1016/J.COMPSTRUCT.2021.114714.
12. Kaware Kiran, Kotambkar Mangesh. Low velocity impact response and influence of parameters to improve the damage resistance of composite structures/materials: a critical review. *Https://DoiOrg/101080/135882652021191 4985* 2021;27(4):1232–56. Doi: 10.1080/13588265.2021.1914985.
13. Çetin Mehmet Emin. The effect of carbon nanotubes modified polyurethane adhesive on the impact behavior of sandwich structures. *Polym Compos* 2021;42(9):4353–65. Doi: 10.1002/PC.26153.
14. Ma Quanjin, Rejab M. R.M., Siregar J. P., Guan Zhongwei. A review of the recent trends on core structures and impact response of sandwich panels. *J Compos Mater* 2021;55(18):2513–55. Doi: 10.1177/0021998321990734.
15. Fakhreddini-Najafabadi Sajjad, Torabi Mahdi, Taheri-Behrooz Fathollah. An experimental investigation on the

low-velocity impact performance of the CFRP filled with nanoclay. *Aerosp Sci Technol* 2021;116:106858. Doi: 10.1016/J.AST.2021.106858.
16. Hervin F., Maio L., Fromme P. Guided wave scattering at a delamination in a quasi-isotropic composite laminate: Experiment and simulation. *Compos Struct* 2021;275:114406. Doi: 10.1016/J.COMPSTRUCT.2021.114406.
17. Rellinger T., Underhill P. R., Krause T. W., Wowk D. Combining eddy current, thermography and laser scanning to characterize low-velocity impact damage in aerospace composite sandwich panels. *NDT & E International* 2021;120:102421. Doi: 10.1016/J.NDTEINT.2021.102421.
18. James Robin, Joseph Roshan Prakash, Giurgiutiu Victor. Impact Damage Ascertainment in Composite Plates Using In-Situ Acoustic Emission Signal Signature Identification. *Journal of Composites Science 2021, Vol 5, Page 79* 2021;5(3):79. Doi: 10.3390/JCS5030079.
19. Bishop Christopher M. Pattern Recognition And Machine Learning - Springer 2006. *Antimicrob Agents Chemother* 2014;58(12):7250–7.
20. Kurian Bibin, Liyanapathirana Ranjith. Machine Learning Techniques for Structural Health Monitoring. *Lecture Notes in Mechanical Engineering*; 2020. p. 3–24.
21. PERAFÁN-LÓPEZ Juan Carlos, SIERRA-PÉREZ Julián. An unsupervised pattern recognition methodology based on factor analysis and a genetic-DBSCAN algorithm to infer operational conditions from strain measurements in structural applications. *Chinese Journal of Aeronautics* 2021;34(2):165–81. Doi: 10.1016/J.CJA.2020.09.035.
22. Salehi Hadi, Das Saptarshi, Chakrabartty Shantanu, Biswas Subir, Burgueño Rigoberto. Damage identification in aircraft structures with self-powered sensing technology: A machine learning approach. *Struct Control Health Monit* 2018;25(12):e2262. Doi: 10.1002/STC.2262.
23. Ai Li, Soltangharaei Vafa, Bayat Mahmoud, Van Tooren Michel, Ziehl Paul. Detection of impact on aircraft composite structure using machine learning techniques. *Meas Sci Technol* 2021;32(8):084013. Doi: 10.1088/1361-6501/ABE790.
24. Shi Sheng, Du Dongsheng, Mercan Oya, Kalkan Erol, Wang Shuguang. A novel unsupervised real-time damage detection method for structural health monitoring using machine learning. *Struct Control Health Monit* 2022;29(10):e3042. Doi: 10.1002/STC.3042.
25. Wang Xing er, Meng Yanran, Yang Jian, Huang Xuhao, Wang Feiliang, Xu Han. Optimal kernel extreme learning machine model for predicting the fracture state and impact response of laminated glass panels. *Thin-Walled Structures* 2021;162:107541. Doi: 10.1016/J.TWS.2021.107541.
26. Cabrera José M., Rajput Abhishek, Iqbal M. A., Gupta N. K. Performance of various thin concrete slabs under projectile impact: Sobol's sensitivity analysis with aid of metamodels. *Thin-Walled Structures* 2022;172:108739. Doi: 10.1016/J.TWS.2021.108739.
27. Dipietrangelo F., Nicassio F., Scarselli G. Structural Health Monitoring for impact localisation via machine learning. *Mech Syst Signal Process* 2023;183:109621. Doi: 10.1016/J.YMSSP.2022.109621.
28. Piezoelectric Discs. Available at https://www.physikinstrumente.com/en/products/piezoelectric-transducers-actuators/disks-rods-and-cylinders/piezoelectric-discs-1206710/#downloads. Accessed March 1, 2022, n.d.
29. A.S.S.E. Lab. Available at https://asselab.unisalento.it/en/. Accessed March 1, 2022, n.d.
30. Pico Technology. Available at https://www.picotech.com/. Accessed May 1, 2023, n.d.
31. Santoni Giola B., Yu Lingyu, Xu Buli, Giurgiutiu Victor. Lamb Wave-Mode Tuning of Piezoelectric Wafer Active Sensors for Structural Health Monitoring. *J Vib Acoust* 2007;129(6):752–62. Doi: 10.1115/1.2748469.
32. Carrino S., Maffezzoli A., Scarselli G. Active SHM for composite pipes using piezoelectric sensors. *Mater Today Proc* 2021;34:1–9. Doi: 10.1016/J.MATPR.2019.12.048.
33. Schindler Paul M., May Russell G., Claus Richard O., Shaw J. K. Location of impacts on composite panels by embedded fiber optic sensors and neural network processing. *Smart Structures and Materials 1995: Smart Sensing, Processing, and Instrumentation* 1995;2444:481–9. Doi: 10.1117/12.207698.
34. Raspberry Pi. Available at https://www.raspberrypi.com/. Accessed May 1, 2023, n.d.
35. BitScope test, measurement & data acquisition. Available at https://www.bitscope.com/. Accessed May 1, 2023, n.d.
36. Paluszek Michael, Thomas Stephanie. MATLAB machine learning recipes: A problem-solution approach, Second edition. *MATLAB Machine Learning Recipes: A Problem-Solution Approach, Second Edition* 2019:1–347. Doi: 10.1007/978-1-4842-3916-2/COVER.
37. Glorot Xavier, Bengio Yoshua. Understanding the difficulty of training deep feedforward neural networks n.d.
38. Kayri Murat. Predictive Abilities of Bayesian Regularization and Levenberg–Marquardt Algorithms in Artificial Neural Networks: A Comparative Empirical Study on Social Data. *Mathematical and Computational Applications 2016, Vol 21, Page 20* 2016;21(2):20. Doi: 10.3390/MCA21020020.
39. Møller Martin Fodslette. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks* 1993;6(4):525–33. Doi: 10.1016/S0893-6080(05)80056-5.
40. De Luca Alessandro, Perfetto Donato, Lamanna Giuseppe, Aversano Antonio, Caputo Francesco. Numerical

Investigation on Guided Waves Dispersion and Scattering Phenomena in Stiffened Panels. *Materials 2022, Vol 15, Page 74* 2021;15(1):74. Doi: 10.3390/MA15010074.