



Science Arts & Métiers (SAM)

is an open access repository that collects the work of Arts et Métiers Institute of Technology researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: <https://sam.ensam.eu>
Handle ID: <http://hdl.handle.net/10985/23342>

To cite this version :

Hadrien POSTORINO, Eric MONTEIRO, Marc RÉBILLAT, Nazih MECHBAL - Experimental Damage Localization and Quantification with a Numerically Trained Convolutional Neural Network - In: EWSHM2022 (European Workshop on Structural Health Monitoring) 4-7 July 2022, Italie, 2022-07-04 - EWSHM2022 - 2022

Any correspondence concerning this service should be sent to the repository

Administrator : scienceouverte@ensam.eu



Experimental damage localization and quantification with a numerically trained convolutional neural network.

Hadrien Postorino¹, Eric Monteiro¹, Marc Rebillat¹ and Nazih Mechbal¹

¹ Laboratoire PIMM, Arts et Metiers Institute of Technology, CNRS, Cnam, HESAM Université, 151 Boulevard de l'Hôpital, 75013 Paris, France
hadrien.postorno@ensam.eu

Abstract. Structural Health Monitoring (SHM) based on Lamb wave propagation is a promising technology to optimize maintenance costs, enlarge service life and improve safety of aircrafts. A large quantity of data is collected during all the life cycle of the structure under monitoring and must be analysed in real time. We propose here to use 1D-CNN to estimate the severity and the localisation of a damage with the signals measured on a composite structure monitored with piezoelectric transducers (PZT). Two architectures have been tested: one takes for input the difference of the time signals of two different states and the second takes for inputs temporal damage indexes. Those simple networks with a few layers predict with high precision the position and the severity of a damage in a composite plate. The evaluations on different cases show the robustness to simulated manufacturing uncertainties and noise. An evaluation on experimental measurement shows promising results to localise a damage on a real plate with a CNN trained with numerical data.

Keywords: SHM, Lamb Wave, Deep Learning, CNN, Damage localization, Damage quantification.

1 Introduction

Structural Health Monitoring (SHM) is a promising technology to optimize maintenance costs, enlarge service life and improve safety of aircrafts. In SHM systems dedicated to the monitoring of aeronautic composite parts and based on Lamb Wave (LW) propagation, damage detection, localization, and size estimation rely on the post-processing of LW signals measured by a network of PZT acting alternatively as actuator or sensor [1]. Ensuring the full reliability of these tasks among a full aircraft fleet is extremely challenging due to their complex dynamical behaviors and to inherent manufacturing uncertainties, as well as different environmental and operational contexts.

The paradigm of SHM lay on the collection of large amounts of data during all the life cycle of a structure under monitoring. The use of methods able to analyse the

huge quantity of data collected during a SHM interrogation thus become necessary. Deep Learning strategies and especially Convolutional Neural Network (CNN) have shown their capability to extract features from related data, especially for image recognition or time-series classification, with good generalization performance [2]. Different architectures of network have been proposed other the last years, improving considerably the performance of the networks. One of them, the network VGG16 proposed by Simonyan [3] shows excellent results for image classification with a relatively simple design. These capabilities can be useful in SHM to characterize damage state with manufacturing uncertainties or varying environmental conditions. We focus here on the localisation and the quantification of a damage.

In the field of SHM, Deep Learning has become more and more popular over the last few years and some application in LW SHM got interesting results. Ewald [4] was one of the firsts to use a CNN in SHM for the classification of damages scenario on an aluminium plate. Rautela [5] compares the performance of a 1D-CNN, 2D-CNN and a Long Short Term Memory (LSTM) network for the detection and localization of damages. The author shows that the 1D-CNN has the best results and outperforms the classical machine learning approaches. Tabian [6] uses 2 CNN to estimate the position and the energy level of an impact on an experimental aeronautic structure. Zhang [7] uses a Multi-Tasks CNN with 3 tasks (estimation of position x , position y and damage level) to localize and quantify a damage on an aluminium plate. Zhang [8] introduces Time Varying Damage Index with a 1D-CNN to localize a damage in an experimental aluminium plate.

One of the limits of those approaches is the use of costly experimental data to train the CNN. Numerical data are much easier to get but generally doesn't fit sufficiently with experimental data. This work is a preliminary approach to estimate the possibility of training a CNN with numerical data and use it with experimental data. We propose here to estimate the severity and the position of a damage on a composite plate with 1-dimensional CNN trained with numerically generated data. Two architectures with different inputs are tested. The first one takes for input temporal signal, while the second one takes for input a temporal damage index (TDI). Both architectures are like the VGG16 network but adapted for time-series regression. Then we test the trained network on data with simulated manufacturing uncertainties and noisy measurements to evaluate the generalization possibilities. Finally, the trained networks are used to evaluate the position of a delamination on an experimental real plate.

Table 1. Mechanical properties of the composite plate

Name	E_1	E_2	ν_1	ν_2	ρ	G_1	G_2
Value	64.5	65.5	0.054	0.054	1570	4.8	4.8
Unit	MPa	MPa	-	-	kg/m ³	MPa	MPa

2 Database Computation

We propose here to estimate the severity and the position of a damage on a structure with a 1-dimensional CNN. We obtained the data with Finite Elements (FE) simulations with the software SDTools [9]. The structure studied here is a composite plate ($600 \times 600 \text{ mm}^2$) with a network of 5 PZTs. The mechanical properties of the plate and the model characteristics can be seen Table 1. Those properties have been adjusted to fit as best as possible to experimental measurements. The damage is modelled by a circular reduction of Young and shear moduli. The damage size is fixed at 10mm. Only the position and the severity (the moduli reduction in the 10mm of diameter damage) of the damage vary. To limit the simulation cost, the damages are localized in a polygon delimited by the PZTs. The excitation signal is a burst of 5 cycles with a central frequency of 100 kHz and 10V amplitude. A first database on a structure S1 is generated with 369 damage configurations computed for each actuator, leading to 1845 simulations. 75% of damage configurations are used for training, and 25% kept for testing. A second database with 80 damage configurations on the structure S2 is generated with small random variations (around 2mm) on the PZT positions to simulate manufacturing uncertainties. The CNN is trained with the data of the first database and tested with the second database to estimate the robustness of the network. A third database is computed by adding a gaussian noise to the previously mentioned database. We choose a noise level equal as experimental noise level with a 70dB SNR. To simulate multiple measurements, each original sample is duplicated 5 times and then the noise is added. Last but not least, a fourth experimental database composed of measurements on a real plate damaged using symmetrical laser shock [10] is used to verify if the CNN trained with numerical data can localise the delamination on the real plate. The delamination is a 4mm circle in the middle ply of the composite.

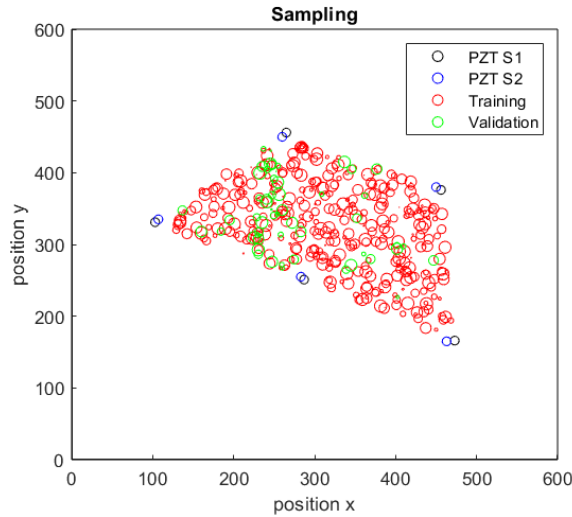


Figure 1. PZT positions and samples for training and validation sets for database 2

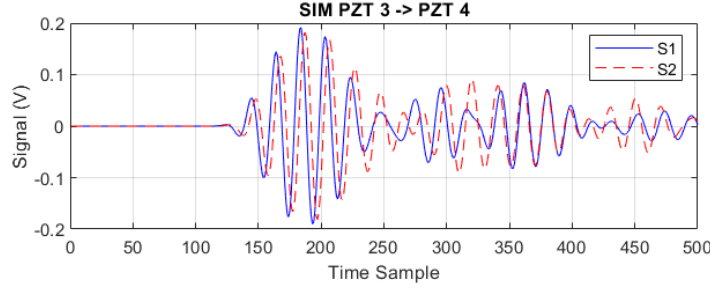


Figure 2. Healthy signal for structure S1 and S2 with actuator 3 and sensor 4

3 Networks architectures

To create a new architecture adapted to the desired task, different kinds of layer must be combined. We use here the toolbox Keras [11] to implement our CNN. A classical CNN contain two main consecutive blocs: a first bloc extracts features from the data and a second bloc uses them to solve the classification or regression problem. The first bloc is commonly composed of successive superpositions of two kinds of layer: convolution layers and pooling layer. The convolution layer compute features from the input of the layer by applying a convolution filter. The number of filter and their size are the main parameters for that layer. The filter itself is computed during the optimization process. At the end of the convolution layer, an activation function such as ReLu or SoftMax is applied to the features adding non-linearity to the network. After the convolution, the pooling layer reduces the size of the features by applying a function such as mean or max on a patch of features. The second bloc of the network is a multi-layer perceptron with a few fully connected layers. This bloc takes for input the features extracted from the original data by the first bloc and solve the classification or regression task.

The optimization of such network is a huge challenge because of the high number of parameters. As example, the VGG-16 network has 138 million parameters to optimize. Different optimization procedure exists such as Stochastic Gradient Decent (SGD) or Adam [12]. They rely on the backpropagation of gradient introduced by Lecun [2]. A lot of possible architectures could solve our problem, we will present here two different architectures with two different inputs.

TIME CNN. The first CNN (table 2) takes for input the difference time signals for each actuator-sensor path concatenated in one vector. This network has a total of 7 layers (5 for the features extraction bloc and 2 for regression bloc). The first convolution layer uses a large filter of size 50 and then a max pooling with the same size reduces the data. We choose here a large filter size because the sampling rate is high (2 MHz) leading to redundant information in the input signal. Then, as suggested the VGG-16 network, we use a small filters size for the next convolutions. The 2nd and 3rd

convolutions apply 16 filters of size 3. The regression bloc is composed of one layer of 128 fully connected neurons and 3 neurons for the outputs.

TDI CNN. To reduce the dimension of the input space and facilitate the training of the CNN, features can be extracted from the LW signals. A classical approach in SHM consist of computing Damage Indexes (DI), giving synthetic information about the damage state. However, most of the DIs do not contain temporal information, impeding localization. We propose here to compute DIs on a rectangle window of size 200 samples translating of half a period along the LW signals to keep information about the damage position. The features vectors obtained for all the pair of actuator-sensor are concatenated and used for training. We choose here the absolute standard deviation as DI. As the TVDI contains condensed information of the signal, we use only two convolution layers with 8 filter and 16 filters of size 3 (table 3). After each convolution, a max pooling layer of size 3 reduces the features size. The regression bloc contains one layer of 64 fully connected neurons and 3 neurons for the outputs. For the optimization of those two networks, we use the Adam algorithm. The loss function is the mean square error. The optimization is computing during 200 epochs with a batch size of 30.

Table 2. Architecture of the Time CNN

Layer type	Activation function	Parameter Name	Parameter value	Output size
Input	-	-	-	12550
Convolution	ReLu	Kernel	50x8	12501x8
Max Pooling	-	Pooling size	50	250x8
Convolution	ReLu	Kernel	3x16	248x16
Max Pooling		Pooling size	3	82x16
Convolution	Relu	Kernel	3x16	80x32
Max Pooling	-	Pooling size	3	26x32
Flatten	-	-	-	832
Dense	Relu	Weight	64	128
Output	-	Weight	3	3

Table 3. Architecture of TDI-CNN

Layer type	Activation Function	Parameters names	Parameters value	Output size
Input				251
Convolution	ReLu	Kernel	8x3	249x16
Max Pooling		Pooling size	2	124x16
Convolution	Relu	Kernel	16x3	122x32
Max Pooling		Pooling size	2	61x32
Flatten				1952
Dense	Relu	Weight	128	128
Output		Weight	3	3

4 Networks evaluation

At first, the two networks are trained with the database 1. 75% of that database is used for training and 25% used for testing. The testing data has not been seen during the training process. As we can see on figure 3, the two networks got good results for the localization and severity estimation of the damage. The Time-CNN got slightly better results than the TDI-CNN for the localization. It is the opposite concerning the severity estimation. Those results could be explained by the averaging properties of the TDI computation. Then, we use 100% of the database 1 to train the networks and we test them with the database 2. The small variations of the PZTs positions generate signals with important differences (figure 2) because the group velocity of LW is high in our material. In that case either, the two networks show similar performances. Then, the noisy dataset is used to train the network. The mean of the error is not significantly influenced by the noise but the variance of the error increase considerably (figure 3). Surprisingly, the severity estimated by the networks presents better mean error with noisy data, but the variance is very high, meaning that the confidence of this predictions is low. Finally, we use the previously trained CNNs to predict the position on a real damaged plate. The result of the figure 3 suggests that our network can find the zone of the damage, the localisation error is around 25mm for the two architectures. More tests must be conducted on different damage positions and size to validate our approach.

5 CONCLUSION

We have investigated here the possibility to use CNN for solving two major steps of the SHM process: estimate the size and the position of a damage. Two architectures have been tested: one takes for input the difference of the time signals for two different states and the second takes for inputs temporal damage indexes. Those simple CNN with a few layers can predict with high precision the position and the severity of

a damage in a composite plate. This network seems even robust to some manufacturing uncertainties and noise measurement. It is therefore conceivable to test those trained networks on real experimental data. The differences between the model and the experimental plate are strong enough to reduce the precision of the networks as shown the results on experimental data. To face that issue, the use of transfer learning methods could be interesting.

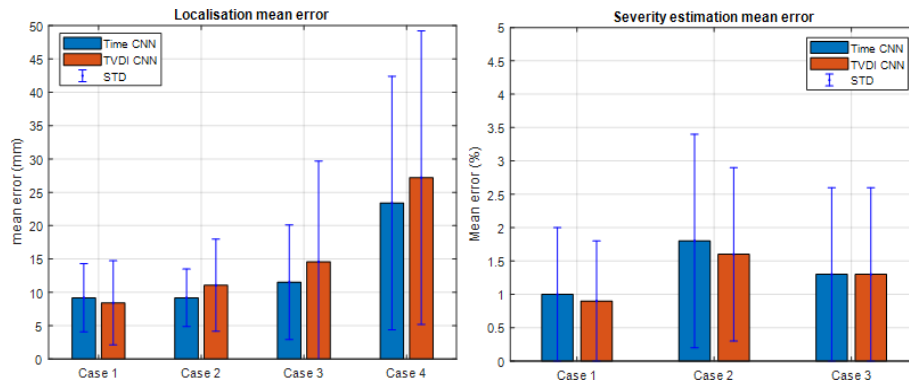


Figure 3. Evaluation of the two networks on the 4 cases

References

1. U. Gabbert, R. Lammering, T. Schuster, M. Sinapius, et P. Wierach, Éd., *Lamb-Wave Based Structural Health Monitoring in Polymer Composites*, 1st ed. 2018. Cham: Springer International Publishing : Imprint: Springer, 2018. doi: 10.1007/978-3-319-49715-0.
2. Y. LeCun, Y. Bengio, et G. Hinton, « Deep learning », *Nature*, vol. 521, n° 7553, p. 436-444, mai 2015, doi: 10.1038/nature14539.
3. K. Simonyan, A. Vedaldi, et A. Zisserman, « Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps », *ArXiv13126034 Cs*, avr. 2014, Consulté le: avr. 21, 2021. [En ligne]. Disponible sur: <http://arxiv.org/abs/1312.6034>
4. V. Ewald, R. Groves, et R. Benedictus, *DeepSHM: A Deep Learning Approach for Structural Health Monitoring Based on Guided Lamb Wave Techniques*. 2019. doi: 10.1117/12.2506794.
5. M. Rautela et S. Gopalakrishnan, « Ultrasonic guided wave based structural damage detection and localization using model assisted convolutional and recurrent neural networks », *Expert Syst. Appl.*, vol. 167, p. 114189, avr. 2021, doi: 10.1016/j.eswa.2020.114189.
6. I. Tabian, H. Fu, et Z. S. Khodaei, « A Convolutional Neural Network for Impact Detection and Characterization of Complex Composite Structures », *Sensors*, vol. 19, n° 22, p. 4933, nov. 2019, doi: 10.3390/s19224933.

7. B. Zhang, X. Hong, et Y. Liu, « Multi-Task Deep Transfer Learning Method for Guided Wave-Based Integrated Health Monitoring Using Piezoelectric Transducers », *IEEE Sens. J.*, vol. 20, n° 23, p. 14391-14400, déc. 2020, doi: 10.1109/JSEN.2020.3009194.
8. S. Zhang, C. M. Li, et W. Ye, « Damage localization in plate-like structures using time-varying feature and one-dimensional convolutional neural network », *Mech. Syst. Signal Process.*, vol. 147, p. 107107, janv. 2021, doi: 10.1016/j.ymssp.2020.107107.
9. E. Balmes et A. Deraemaeker, *Modeling structures with piezoelectric materials. Theory and SDT tutorial*. SDTools, Paris, France, 2013. [En ligne]. Disponible sur: <http://www.sdtools.com/help/piezo.pdf>
10. M. GHRIB, « Structural Health Monitoring of composite structures : LASER shock delamination generation and machine learning-based quantification », Theses, Ecole nationale supérieure d'arts et métiers - ENSAM, 2017. [En ligne]. Disponible sur: <https://pastel.archives-ouvertes.fr/tel-01881284>
11. F. Chollet et others, « Keras ». 2015. [En ligne]. Disponible sur: <https://keras.io>
12. D. P. Kingma et J. Ba, « Adam: A Method for Stochastic Optimization », *ArXiv14126980 Cs*, janv. 2017, Consulté le: mai 19, 2021. [En ligne]. Disponible sur: <http://arxiv.org/abs/1412.6980>