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/10985/25141

To cite this version :

Arthur LEJEUNE, Nicolas HASCOËT, Eric MONTEIRO, Nazih MECHBAL, Marc RÉBILLAT - An enhanced topological analysis for Lamb waves based SHM methods - Structural Health Monitoring p.1-21 - 2023

Any correspondence concerning this service should be sent to the repository Administrator : scienceouverte@ensam.eu

Arts et Métiers Institute of Technology

An enhanced topological analysis for Lamb waves based SHM methods

Arthur Lejeune[1,2], Nicolas Hascoët[1], Marc Rébillat[1], Eric Monteiro[1] and Nazih Mechbal[1]

Abstract

Topological data analysis (TDA) is a powerful and promising tool for data analysis, but yet not exploited enough. It is a multidimensional method which can extract the topological features contained in a given dataset. An original TDA-based method allowing to monitor the health of structures when equipped with piezoelectric transducers (PZTs) is introduced here. Using a Lamb wave based Structural Health Monitoring (SHM) approach, it is shown that with specific pre-processing of the measured time-series data, the TDA (persistent homology) for damage detection and classification can be greatly improved. The TDA tool is first applied directly in a traditional manner in order to use homology classes to assess damage. After that, another method based on slicing the temporal data is developed to improve the persistence homology perception and to leverage topological descriptors to discriminate different damages. The dataset used to apply both methods comes from experimental campaigns performed on aeronautical composite plates with embedded PZTs where different damage types have been investigated such as delamination, impacts and stiffness reduction. The proposed approach enables to consider a priori physical information and provides a better way to classify damages than the traditional TDA approach. In summary, this article demonstrates that manipulating the topological the features of PZTs time-series signals using TDA provides an efficient mean to separate and classify the damage natures and opens the way for further developments on the use of TDA in SHM.

Keywords

Structural Health Monitoring, experimental data, topological data analysis, time series, damage detection, damage nature classification, supervised learning, neural network, composite aeronautical structures

Introduction

When trying to process and analyse complex information, it is often difficult to exploit raw data. Most technics dedicated to prediction, detection and data-based decision-making need to rely on special structuring and pre-processing steps in order to give better results. A massive raw dataset should not be processed as a single block. In fact, the raw data contain a whole set of information, some of them are useful to predict or detect specific behaviour, but other ones are simply useless in the current application or are repeating some knowledge already present somewhere else in the database. Therefore, the first step of any data analysis process remains to extract useful information in order to fit the dataset to the main purpose of the analysis.

In this article, the main purpose of the analysis is to detect and characterize a damage inside a composite plate structure. In the context of Structural Health Monitoring (SHM), detecting damages and impacts is very important to estimate the remaining useful life[1] and to develop a predictive maintenance process.[2] There are classically five main steps in a SHM process: (1) damage detection, (2) damage localization, (3) damage classification by type, (4) damage severity for the structure and (5) prognostic of the structure health. As detailed in the paper[3] written by Rahbari et al. the SHM research community agreed on solving these issues through four different steps. First a physical evaluation of the health structure, then acquisition of data measured by sensors to provide information about the structure. After that, it is necessary to extract features

[1]PIMM Laboratory, Arts et Métiers Institute of Technology, CNRS, CNAM, HESAM University, Paris, France
[2]Safran Composites, A Technology Platform of Safran Tech, Itteville, France

Corresponding author:
Arthur Lejeune, PIMM Laboratory, Arts et Métiers Institute of Technology, CNRS, CNAM, HESAM University, 151 Boulevard de l'Hopital, Paris 75013, France.
Email: arthur.lejeune@ensam.eu

that underlines the structure health to finally train models using damage features and real health indicators as input data. Among all kinds of existing SHM techniques, the ultrasonic guided waves strategy is particularly effective for such composite structures because guided waves can propagate over important distances and thus cover a large area with few sensors and reduced testing times.[4] There exist many ways to extract relevant information from data using time series, statistical approaches or Fourier transform, for example. These approaches are well covered by the literature.[1,2,5] Many methods have been validated to assess several steps for SHM purposes. Wavelet analysis with Lamb wave measurement can be efficient for quantification.[4] Detection and classification of damage on composite structures were solved thanks to order reduction methods such principal component analysis or more complex algorithms.[6,7] Two different approaches can be found to estimate the health of a structure, the first one is supervised learning or modelling, which need the physical control result to work. On the other hand, unsupervised learning aims to detect, localize or cluster with the embedded sensor data alone.[6–8] In SHM, synthetic damage indicators build up from time series and called damage index are used instead of raw measurements. Damage indexes can be built upon the understanding of the physical behaviours of propagating waves. For example, the time of travel of the waves or the residual energy can be computed from temporal signals. The spectral analysis with FFT gives frequential energy, critical modes. Wavelet analysis can also provide several coefficients to enrich the damage index list. They can also be built arbitrarily by using some empirical mathematical formulae. Methods relying on such damage indexes have already proved to be able to monitor damages inside a composite structure.[2,5,9]

The topological data analysis (TDA) is an alternative method that allows to extract topological features from a given dataset. It can be interpreted as an original manner to quantify the '*shape*' of multidimensional data and provides a new way to infer damage indexes for high dimensional data spaces. The main algebraic topology tool used in the TDA is called *persistence homology*. The *persistence homology* gives information about the global and local form of the data and consequently extracts topological invariant features from the data. Previous research works have proven the efficiency of the topological analysis in several domains. In financial networks, topological methods are used to detect critical transition in financial data.[10] TDA is also useful in medicine[11–13] to analyse functional magnetic resonance imaging data of the brain.[14] Topological analysis enables to classify human body movements during activities.[15] An introductive and first attempt, well-documented work, on the use of TDA for SHM is given in Gowdridge's article[16]. In the previously mentioned article, an approach based on the concept of persistent homology, which represents a general mathematical framework (see the Appendix for more details on the *homology* concept) able to encode the evolution of the topology (homology) of families of nested spaces has been explored to analyse damage on a mass-spring-damper model and a Z24 bridge case study. The approach used a barcode persistence representation[17,18] assessing the *Betti number* evolution and the *p-Wasserstein distance*[19] to exhibit the effect of damages, in terms of manifold structures, on the data topology.

In this article, an extension of these pioneering works is proposed to tackle the problem of damage monitoring in composite structures using guided wave-based techniques. The article will show that a direct application of the TDA on the raw measured signals is not effective to classify damages. However, by slicing the temporal data into several time-windows, it is possible to highlight similarities existing between each damage type and then to successfully classify them. In fact, each window generates unique persistence image using the *lower-star filtration* (LSF)[20] method and enable to extract convenient topological features on each data frame. TDA tools enable to represent the extracted topological features in two-dimensional diagrams or images. A simple convolutional neural network (CNN) is a very convenient way to classify the resulting images with few images and few layers. That makes the training step very accurate and fast and could be transferred on many more experimental and simulated data.

Here, four different carbon fibre-reinforced polymer (CFRP) plates endowed with five piezoelectric elements are considered. Three of them have been damaged with different methods trying to reproduce the physical effects of real damage. The SHM approach developed here is based on the analysis of Lamb waves propagation between piezoelectric transducers (PZTs) in both damaged and healthy states. This work falls within the framework of research which tries to bridge the gap between what is undertaken in the laboratory and the industrial deployment of SHM processes.[21] Indeed, the method proposed in this work tackles one of the problems that hinders the industrial deployment of machine learning-based SHM method, which is the strong dependency to the availability of a large database encompassing different damages. In fact, here a small amount of data is necessary to detect and classify damages using the proposed TDA-based approach. A description of the experimental set-up is done in the article to fully understand the context of the dataset, and the physical phenomena that enable to generate and collect data. Then, some TDA for time series

applications and tools are presented. Finally, the TDA tools are applied on SHM experimental data to detect and classify the damage on the composite plate.

Description of the experimental set-up and dataset

The experimental data are provided by PZT bonded on composite CFRP plates. The test specimen is an eight-ply CFRP composite laminate with symmetrical stacking [0°, −45°, +45°, 0°]. The dimensions of each laminate are 400 mm × 300 mm × 0.28 mm. The mechanical properties of the lamina are listed in Table 1, and Figure 1 presents the Lamb wave test bench and the composite specimen with its PZT network.

A set of $N = 5$ PZT elements (Noliac NCE51) from NOLIAC© are surface-mounted on the composite plate.[22] Each piezoelectric element is 20 mm in diameter and 0.1 mm in thickness. The pitch/catch principle is considered here, wherein one PZT acts as an actuator whereas the others act as sensors. The coordinates of the piezoelectric elements are listed in Table 2 and Figure 2.

In addition to the healthy plate, three other similar specimens cut from the same original composite plate have been manufactured and damaged with different types of damage. Each damage is characterized with a position and a radius. As the damages are not produced with the same methods, the physical properties of the damaged area will not be the same. The first type of damage is an impact which is made by dropping masses on the plate (Figure 2). Impacts induce in fact several damages because it breaks down the carbon fibre in addition of the delamination. An illustration of the impact location and of the induced damage is shown in Figure 2. Delamination damage is created, during the manufacturing process, by placing a sheet of polytetrafluoroethylene between two consecutive plies. This material prevents the two layers to join in order to emulate a delamination. The last damage type is a reversible one. It consists of two magnets placed on both side of the plate. The two magnets are cylinders with 30 mm of diameter and has a mass of 150 g each. The magnetic field enables the magnets to stay in place. By pinching the plate, these magnets generate two perturbations on the dynamic behaviour by adding a local mass and by locally modifying the boundary conditions. All these damages try to represent different type of mechanical disturbances that can be met in practice during the composite structure life cycle (Table 3).

In Figure 2(a), the damage is located at (300, 150) mm. The excitation signal is generated by a 33500B Series Waveform Generator (Trueform) and amplified to 10 V using a voltage amplifier from FLC Electronics. It is a five-cycle sinusoidal tone burst with a central frequency of $f_0 = 200$ kHz, modulated by a Hanning window (Morlet wavelet type). The sampling frequency is fixed to 1 MHz.

Successively, each PZT becomes the transmitter and the four other the sensors. The same experiment is repeated 100 times in order to assess the repeatability of the process. On the 'smart composite', it takes only 0.5 s to perform a test. Hence, only 50 s are needed to perform 100 repetitions. At the end, the dataset is composed of 500 time series for each PZT switch from sensor to actuator, multiplied by the four damage types. In the first step, the measured signals are used without any additional post-processing step. A recording in the case of the healthy and delaminated structure is given in Figure 3 as an example.

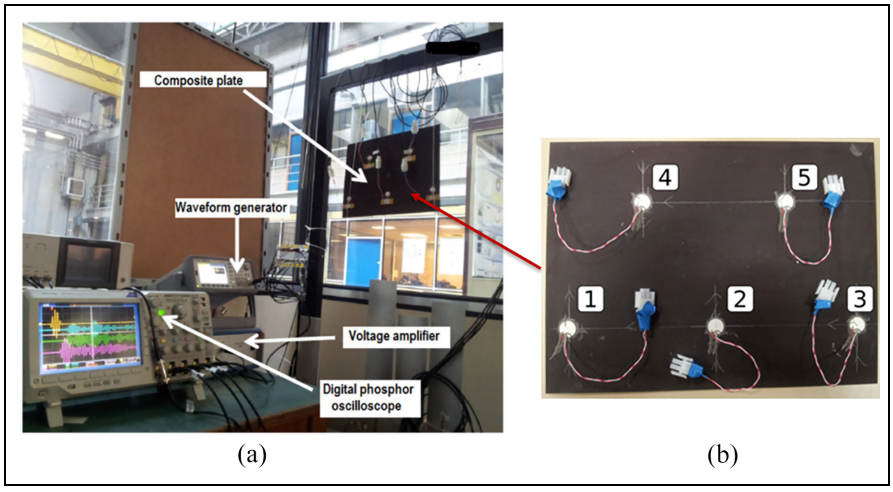

Figure 1. (a) The experimental test bench. (b) The composite plate with the PZT network.
PZT: piezoelectric transducer.

Table 1. Mechanical properties of the composite plate.

Density (kg/m^3)	Ply thickness (mm)	$E_{11} = E_{22}$ (GPa)	E_{33} (GPa)	$G_{12} = G_{13} = G_{23}$ (GPa)	υ_{12}
1554	0.28	69	8.1	4.8	0.03

Table 2. PZT locations.

Coordinate direction	PZT1	PZT2	PZT3	PZT4	PZT5
X (mm)	30	200	370	113.5	286.5
Y (mm)	225	225	225	75	75

PZT: piezoelectric transducer.

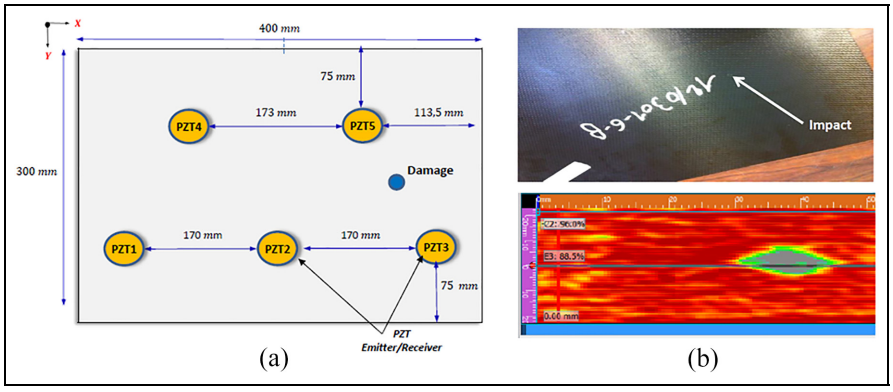

Figure 2. (a) PZT and damage position on the experimental plate. (b) A 14 mm impact (top) C-scan of the impact (bottom).
PZT: piezoelectric transducer.

Table 3. Damage locations and sizes.

Coordinate direction	Impact	Delamination	Magnets
X (mm)	300	100	185
Y (mm)	150	147	155
Radius (mm)	14	10	30

Each sensor measures different signals because they are not located at the same position (cf. Figure 3). In Figure 3, the PZT3 is the actuator, this way the PZT3 signal corresponds to the emitted wave which is the Morlet wavelet mentioned earlier. The first detected packet corresponds to the direct path between the actuator and the sensor. After that, the received signal corresponds to a mix between indirect path waves, which are due to the boundary and to the damage.

It is then necessary to analyse the recorded signals by extracting some features allowing to separate and to classify the different damages. Each damage has a different response to the input signal. That can usually be seen by watching the differences between the damaged responses and the healthy one which is taken as a reference. By comparing the results, it is visible that delamination and impact damages return echoes with more energy than magnets perturbation. That can be explained because the magnets are not damaging the material inside when impact and delamination are doing it. This effect is well-represented in Figure 4, as it can be seen that in fact the difference signal amplitude is lower for the magnets case than other damages. The Lamb waves difference signal will furthermore not be impacted the same way for all the PZT paths. The signal strongly depends on the path followed by the Lamb waves. This is why it is not possible to compare signals from two different paths. In the article, data are always compared path by path to avoid this issue. Watching the signals for one path shows that the magnet damage induces a less energetic footprint than the two other damages. It is also important to mention that the signal depends on the location of the damage because of the reflections of the waves on the borders.

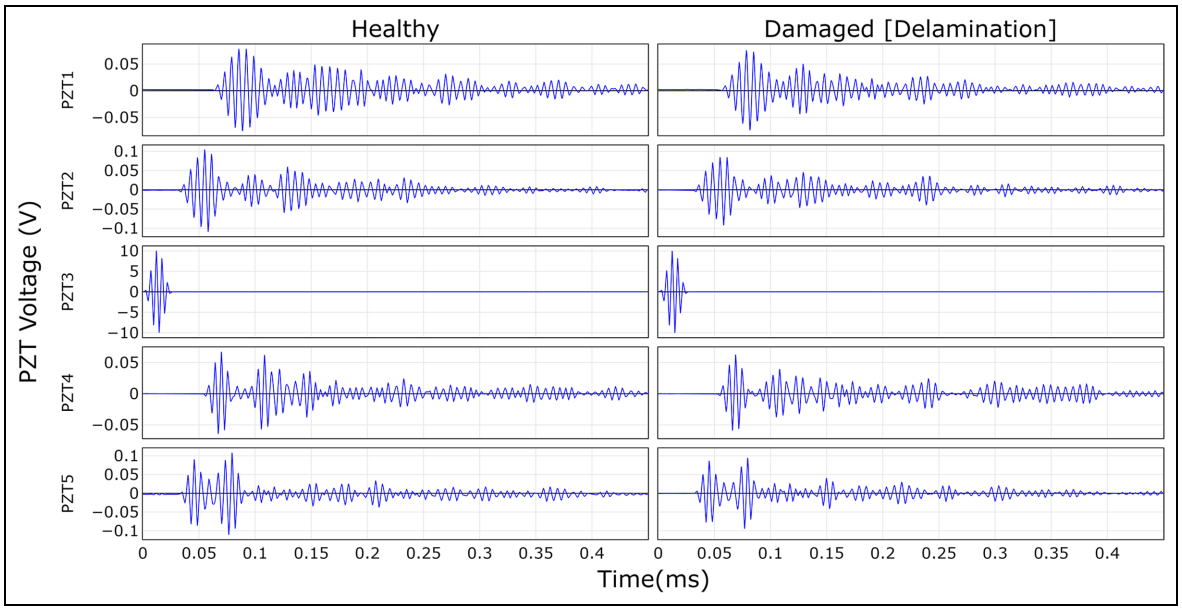

Figure 3. Generated and received signals with PZT3 as actuator for healthy and delaminated plates.
PZT: piezoelectric transducer.

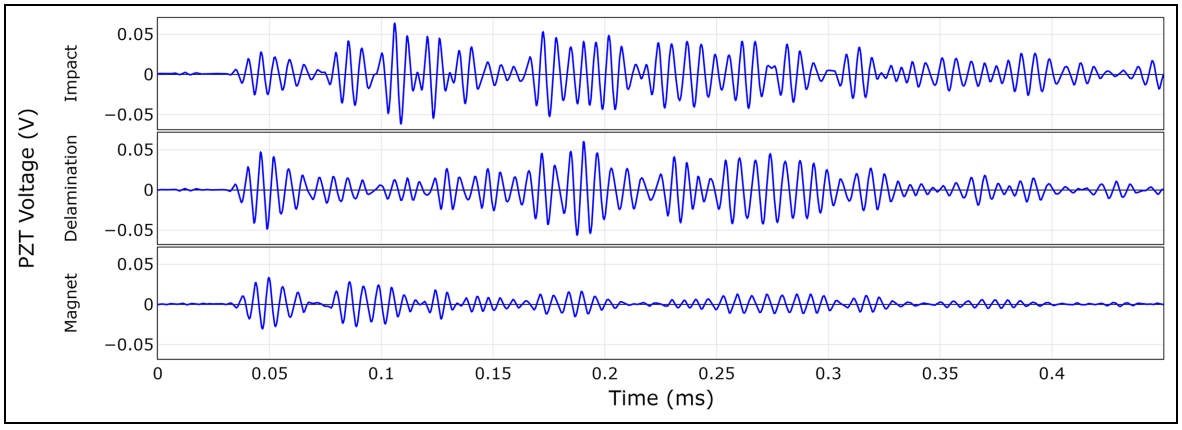

Figure 4. Signal difference between damaged and healthy plates for each damage from actuator 5 to sensor 4.

TDA and results representation

TDA

In this section, the main steps of the TDA are briefly presented. However, for the completeness of the article, all the notions and algorithms quoted here are detailed in the Appendix.

Processed data always has a discrete representation. The first step to achieve TDA is to construct a simplicial complex from the data. It describes the data as a global structure. Processing two-dimensionnal (or more) data is the main purpose for the TDA research works. Algorithms are working the same way regardless of the dimension (as long as it is higher than one). More recently, time series and more generally one-dimensional data analysis has provided interesting results. Several tools have been developed to make it easier.[15,23] Manipulating time series requires to apply extra steps. The main TDA tools are presented on two-dimensional data in Appendix. Several frameworks have been developed for TDA such as Giotto-TDA[24] or Ripser (scikit-tda).[17]

TDA for time series

The major difference between multidimensional data analysis and time series analysis is that there is no defined trivial distance in the one-dimensional space. Time series vertices are sorted along the time axis. To create the simplicial complex and use it for TDA, it is

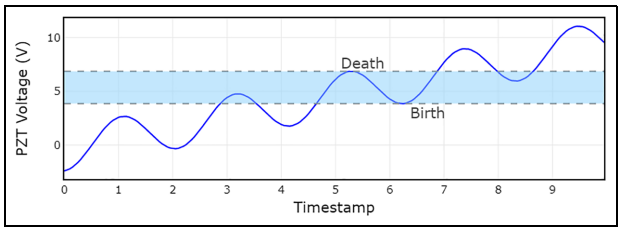

Figure 5. Star lower filtration.

necessary to define such a distance. This distance will provide a way to find neighbours for each point in the time series.

LSF for time series. The LSF,[25] also called *sublevel sets filtration*, describes the local extreme values (minimum and maximum) on the time series. The LSF has an intuitive graphic representation. Homology classes follow a rising level of water on the Y-axis starting at the minimum to the maximum amplitude of the time series. When the water's threshold exceeds a local minimum, a new pool forms, it corresponds to a homology class birth. When the water's level reaches a local maximum and two different pools are merging, it makes the latest homology class die. By considering a raising of the level from the minimum value continuously increasing to the maximum, it gives persistence homology on the dataset. The $t=0$ axis and $t=t_{\max}$ are considered as walls with infinite height (Figure 5).

Let f be a monotonic function $f : K \to \mathbb{R}$ defined on a given simplicial complex K. Being monotonic means if $\Delta_1 \subseteq \Delta_2 \in K$ then $f(\Delta_1) \leqslant f(\Delta_2)$. Applied to mono dimensional time series, the result is that if $t_1 \leqslant t_2$, $f^{-1}(t_1)$ is a subcomplex of $f^{-1}(t_2)$. This means that it is possible to make a sublevel sets filtration with different f^{-1} function. The Vietoris–Rips filtration is the sublevel filtration of the diameter of a circle (cf. Appendix). In the LSF, the increasing threshold is used as the f^{-1} function. The lower-star time filtration approach uses the sparse distance matrix. It is comparable to an adjacency matrix. To build such a matrix, each point is connected to its next neighbours (following time axis). In fact, in a time series, each point has 1 or 2 neighbours. Edges in the simplicial complex are bidirectional that implies the matrix is symmetrical. The sparse distance matrix is reduced to an upper triangular matrix as explained in the two papers written by Bauer[17] and Boissonat et al.[26] After computing the matrix, the Vietoris–Rips filtration can be applied, and the approach become identical two-dimensional method. The persistence diagram obtained only contains H0 homology because the data set is one dimensional, this means that the persistence homology will only be zero dimensional. This way of analysing the data gives information about the range of multi-scale behaviours.

Results representation

The first topology invariant computed is the *Betti number*[27] which gives information about the number and the dimension of appearing homology. There is much more efficient representation of persistence homology. This article will focus on the main tools being used and is not extensive.

Persistence diagrams and persistence images. The persistence diagram is a two-dimensional scatter chart with homology class *birth* as x-axis and *deaths* as y-axis. The birth value is always inferior to death, this implies that all points are situated above the $y=x$ function. All homology classes are represented on the same graph. Dealing with one-dimensional data implies to represent homology of dimensions 0 only as H0 points.

Figure 6 shows the homology persistence of the double cosines time series. First, the nearest the point is from the $(y=x)$ asymptote the less the persistence homology is. Points around the identity function can be considered as noise representation or small amplitude signal because their death is very close to their birth. It is possible to denoise the diagram by filtering those points to get better results. In the diagram of Figure 6, the three highest persistence homologies correspond to the three periods of the high-amplitude cosines signal. The other points are reference to the lower amplitude cosines signal. With this diagram, we can conclude that the signal looks like three 'pools' on which we added some lower amplitude signals with a fixed periodicity.

The *persistence image* is another way to visualize the homology persistence. This image is built in the persistence-death plan as in Figure 7. By using a discretized space on the persistence-death plan it is possible to generate an image from the persistence diagram. This discrete image is used after a convolution with a Gaussian kernel as the persistence image. The computing of the persistence image[23,35] is detailed in the Appendix. Please refer to this Appendix where all the steps and the correspondent equations are summarized.

These persistence images are efficient ways to compare, find similarities and classify datasets. The standard deviation of the Gaussian kernel and the number of pixels of the image are important parameters to generate the image from the original diagram. It also enables to represent the data homology in a vectorial space in order to use Euclidian metrics. These metrics are traditionally preferred for machine learning and image processing operations.

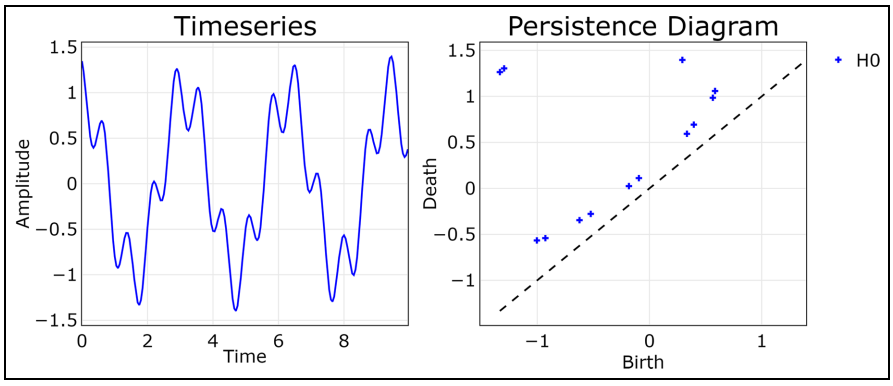

Figure 6. Persistence diagram computed for sum of two cosines signal: $f(t) = \cos(2x) + 0.4\cos\left(8.6x + \frac{\pi}{6}\right)$.

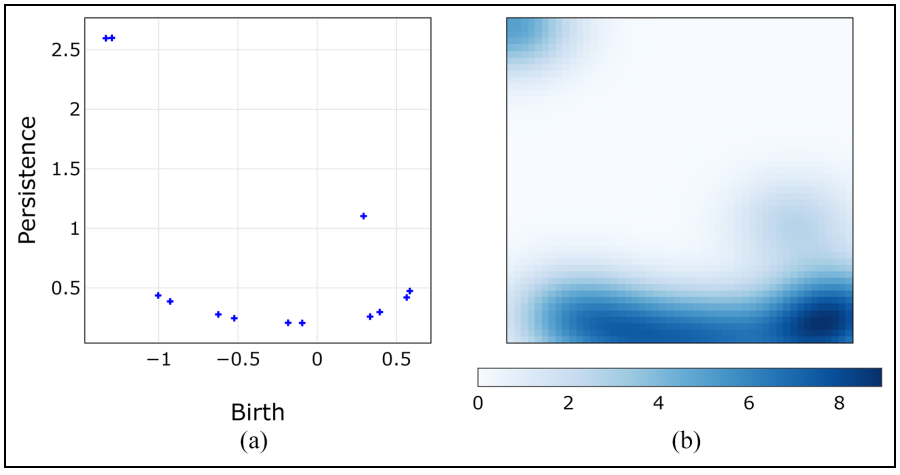

Figure 7. Persistence diagram and image from the two cosines signal: $f(t) = \cos(2x) + 0.4\cos\left(8.6x + \frac{\pi}{6}\right)$. (a) Persistence diagram of H_0 homologies. (b) Persistence image of H_0 homologies.

Diagrams comparison tools. The main purpose for the TDA is to extract topological features in order to compare signals one with another. With image analysis algorithms, it is possible to compare directly or indirectly persistence images. Another method to compare signals with persistence homology is to define a distance between two *persistence diagrams*. The Bottleneck and the Wasserstein distances compute a pairwise distance between points in the two diagrams.

Definition 1: Let D_1 and D_2 be two persistence diagrams, $p>0$, the *p-Wasserstein distance* is:

$$W_p(D_1, D_2) = \inf_{\varphi} \sum_{x \in D_1} ||x - \varphi(x)||_{\infty}^{p} \Bigg)^{\frac{1}{p}} \qquad (1)$$

where $\varphi : D_1 \to D_2$ is the function which gives the pairwise point of $x \in D_1$ in the diagram D_2 and $||x||_{\infty} = \max\{|x_1|, \ldots, |x_n|\}$ then $||x-y||_{\infty} = \max\{|x_1 - y_1|, |x_2 - y_2|\}$ for the two point $x = (x_1, x_2)$ and $y = (y_1, y_2)$.

Definition 2: Similarly, the *Bottleneck distance* is:

$$W_{\infty}(D_1, D_2) = \inf_{\varphi} \left[\sup_{x \in D_1} ||x - \varphi(x)||_{\infty} \right] \qquad (2)$$

The function φ is linked with the *transportation theory* because it links points in pairs by minimizing a cost function which represents the distance between the two points.

Image comparison tools. To use the persistence images to cluster or classify damages in the composite plates, it is mandatory to find calculations to compare images with each other. These indicators will enable to estimate the efficiency of the classification model in the next steps. The simplest way to compare two images is to compute the difference values pixel to pixel on every pixel of the images. To do so, it is necessary to relate same shape images. The first, tools used in the article is the mean square error (MSE). The MSE is defined as:

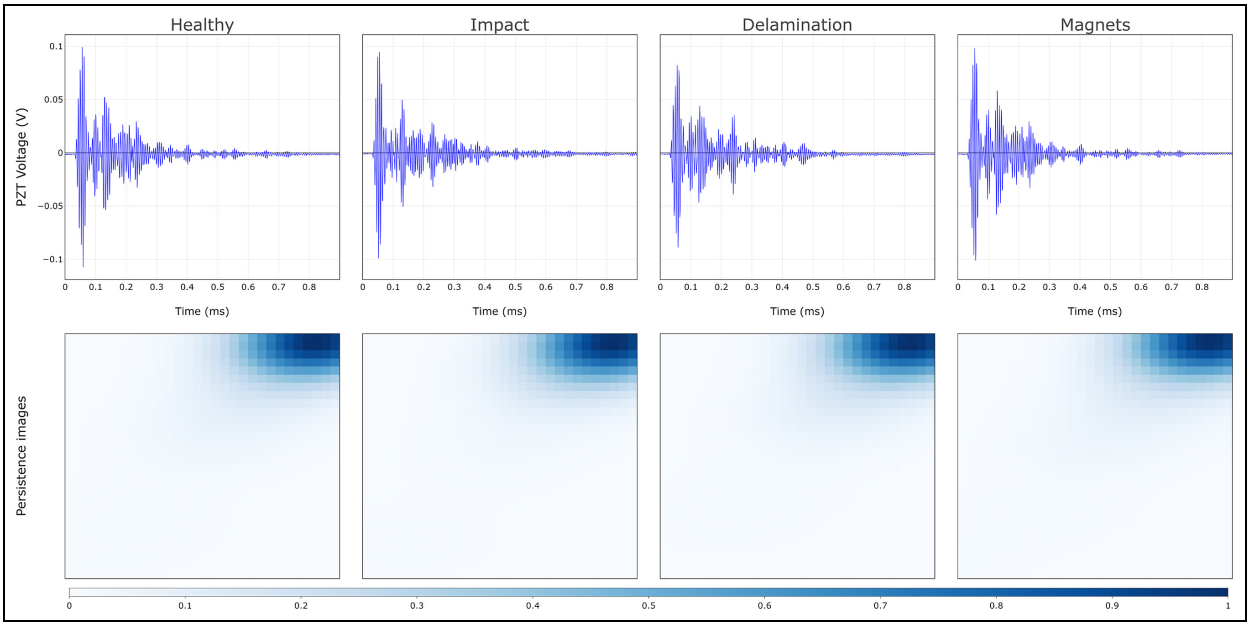

Figure 8. Persistence images for each damage type.

$$MSE(\mathrm{Img}_1, \mathrm{Img}_2) = \frac{1}{N_{\mathrm{pix}}} \sum_{\mathrm{pix}}^{N_{\mathrm{pix}}} ||\mathrm{Img}_1(\mathrm{pix}) - \mathrm{Img}_2(\mathrm{pix})||^2 \tag{3}$$

where Img_1 and Img_2 are the two compared images, and N_{pix} is the number of pixels of the two images.

SHM data analysis application

Applying LSF to time series

As described before, the goal of this work is to use TDA tools to classify data corresponding to different damaged structures and the one coming from healthy plates. Every time series contains the same number of points, these data are sampled at the same sampling frequency. The simplest approach is to perform a LSF on the full time series. Doing so, the obtained result is a single *persistence diagram* for each time series. The Wasserstein distance enables to compare each diagram one to another. With this method, it should be possible to find a difference between healthy experimental data and damaged one. In the remainder of the article, every time series are pre-processed temporal data measured by each sensor. The use of different signals will not be studied here. The main issue of using the whole time series is that the physical propagation of Lamb waves is not considered at all. However, it is known that the received signal is the sum of waves coming from different paths and from boundaries reflexions. The output data from the piezoelectric sensor are strongly related to the input signal and to the time of fly. Unfortunately, the LSF algorithm does not use the temporal information of the data. Therefore, the results of the LSF computed on each time series are not operable for classification or regression purposes.

In Figure 8, it is clearly seen that TDA tools such as LSF on time series without pre-processing and physical understanding cannot be used to classify the damages. In fact, by computing over the whole interval, the LSF only consider highest amplitude data. This leads to ignore lower amplitude data and then miss some useful information. Applying it in a raw way, the LSF applied on time series will not give any good result to detect neither to classify damages.

To cope with this LSF issue, the idea is to split each measured signal into several smaller time series, which can be seen as a windowing processing. Each window will be set at a fixed size. This slicing window size is set to 30 μs for the entire article. It was determined by an impact study of its value on the classification capabilities (cf. 'Impact of the slicing window size' section). It also corresponds to the time duration of the input signal in the experimental set-up. Figure 9 shows the processing pipeline from experimental processing to classification and regression algorithms.

As every time series inside the dataset has the same length, it is possible to compare each window with a constant slicing. The TDA tools do not need aligned time series to successfully compute persistence homology on it. The slicing of each time series into several smaller ones implies to have lined up series in view of

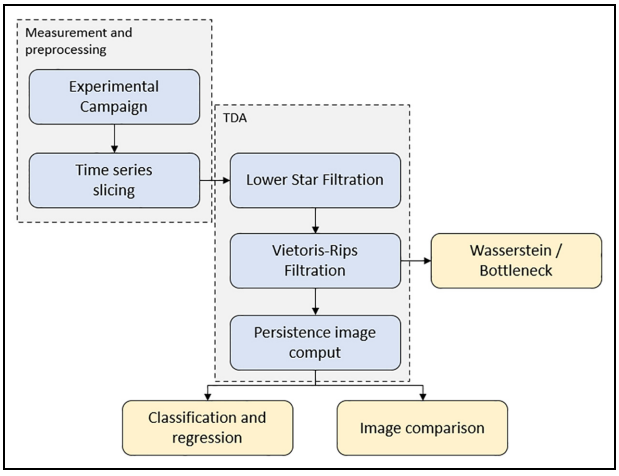

Figure 9. TDA methodology for SHM time series analysis with slicing window method.
TDA: topological data analysis; SHM: Structural Health Monitoring.

the comparison stage between windows. The rephasing operation is done with cross-correlation estimation, which gives the lag between the two signals. The cross-correlation lag is computed for two discrete signals f and g of size N with:

$$(f*g)[l] = \sum_{i=0}^{N} \overline{f[i]}\, g[i+l] \tag{4}$$

where the lag l and $\overline{f[i]}$ are the complex conjugates of $f[i]$.

The chosen lag is equal to the value at the maximum of cross-correlation. Knowing the gap enables to line up every signal on one reference signal. After the slicing, the LSF computes the corresponding persistence diagrams and next step computes the *persistence images*. These images are very convenient to manipulate because of their uniform shapes. On the other side, the quantity of data contained inside every image is bigger than inside the corresponding diagram.

After all these processes, it is possible to construct a dataset of images to work with. Here, for this application, each time series of 900 points length is divided into 30 series of 30 μs. This special value will be discussed in a future section (cf. section 'Impact of the slicing window size'). It is important to notice that, 30 μs corresponds to the duration of the emitted signal. Each window gives one persistence image of 30 pixels by 30 pixels. This means that each measurement time series can be represented as an image of 900 pixels by 30 pixels by concatenating each persistence images from its window time series.

Figure 10 shows an example of the result obtained after applying LSF and Vietoris–Rips filtration on a sliced time series. In figure, the time series is cut on vertical grey lines, which represent the windows. The lower row represents the persistence images computed with the proposed persistence diagrams. The Gaussian kernel is parametrized with the standard deviation σ. This value changes the spreading of each spot on the images. This impact directly the comparison of two images and can improve or deteriorate the classification score. The value of σ is computed with the formula:

$$\sigma(\mathrm{dgm}) = \frac{\max_{h\in \mathrm{dgm}} (\mathrm{persistence}(h))}{n_{\mathrm{pixel}}} \tag{5}$$

In the previous equation, dgm is a persistence diagram containing one or several homology h and n_{pixel} is the number of pixels required in the persistence image. This formula was selected for this study after empirical tests. There is a different sigma value for each computed image. Two diagrams or images taken from the same time series cannot be compared as they do not represent the same window in the data and have different algorithm parameters such as σ. The most interesting thing to do is to compare the result coming from two different time series and corresponding to the same window in the temporal axis. The persistence image generation can fit to each persistence diagram. This makes the persistence image very local information and make the comparison with other persistence images more difficult. In order to compare images coming from different data, it is better to generate them with

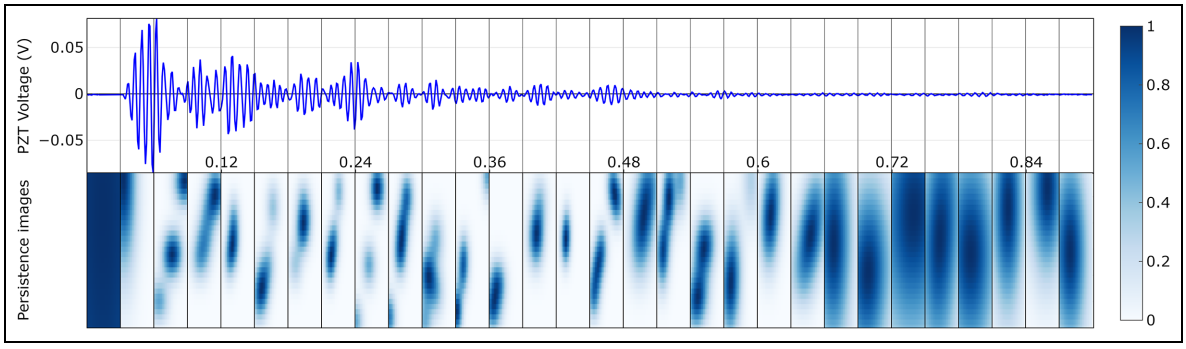

Figure 10. Sliced time series and corresponding persistence diagrams and persistence images for [2–1] path.

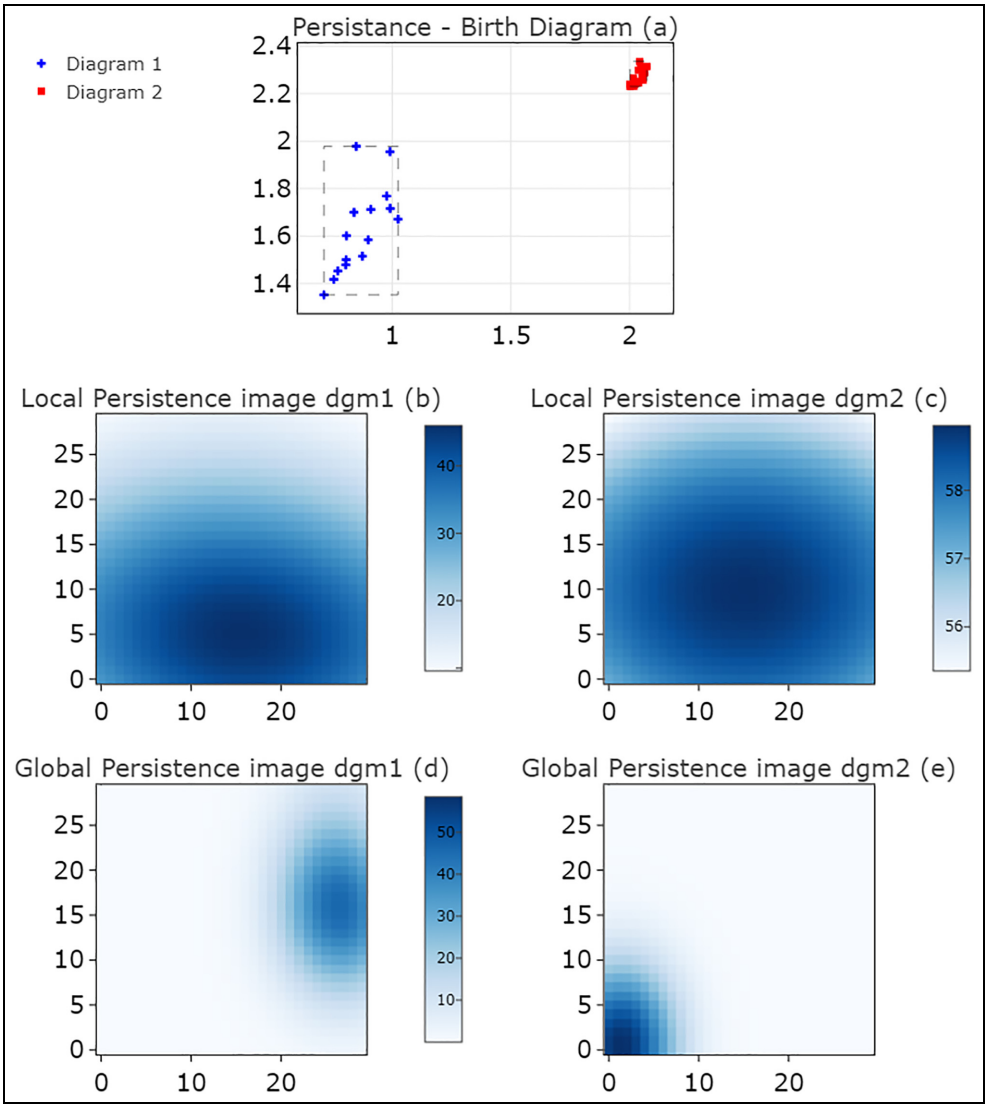

Figure 11. Persistence image generation with global or local fitting parameters.

global fitting parameters. The main parameter to fit is the image range on both axes; by using a global range for image generation, it is possible to compare directly to diagram with their corresponding image. As shown before, σ is the main parameter used to generate persistence images. But there is also a sampling processing which enable to mesh the space and make the Gaussian kernel working. Figure 11 shows the impact on images generated with local or global parameters.

Using global parameters to generate the persistence images gives better results to separate different data. When the persistence images are computed with local parameters, they are very similar as shown in the (b) and (c) images in Figure 11. By taking the full data range and the biggest standard deviation like on the two images (d) and (e) in Figure 11, the two images are very different, and it is easier to compare them. The counterpart of using the global parameters appears when there are more than two diagrams to compare. By increasing the global range of persistence, the lower persistence points will be moving to the top and right of the image. This means that it will be more difficult to differentiate low persistence homology when there are higher persistence homology classes in the same image generation. This issue could be avoided by generating multiple images with different range of parameters. This way it could be possible to use different scaling factors and enable to compare and classifies several diagrams.

As the experimental processes use five PZTs with precise location, the measured waves are strongly dependent of the selected path. The attenuation has a bigger impact on longer paths and damages that change the behaviour of the wave depending on its

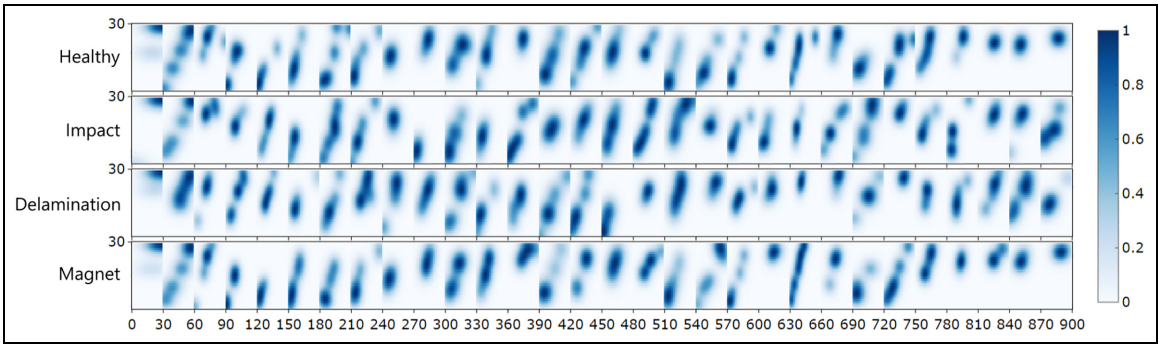

Figure 12. Persistence images for each damage sample.

location compared to the actuator and the sensor. Therefore, each image generation parameter is settled for each different path. Another physical property of the experimental environment is that two opposite paths should provide a very similar response. For example, $\text{actuator}_0 \rightarrow \text{sensor}_1$ and $\text{actuator}_1 \rightarrow \text{sensor}_0$ should be the same. To consider all these rules, σ is computed as follows:

$$\sigma(\text{path, damage, window}) = \max_{\text{direction}} \left(\max_{\text{repetition}} \sigma_{\text{direction, repetition}}(\text{path, damage, window}) \right) \quad (6)$$

Persistence image comparison

Now, the dataset is composed of images (size 900×30 pixels) which represent the topological features of the time series from the experimental dataset.

Figure 12 displays the computed persistence images for one instance of each damage types. The persistence images generated for each sample have common and distinct spots. By analysing and comparing the position and width of each spot, it should be possible to classify the different damages. One can notice that the differences are now visible to the naked eye, which is promising for an automatic classification of the different images. The objective here is to find out differences between healthy and damaged persistence images. To do so, the MSE is used to estimate the distance between each persistence images. These computations give a score to characterize the similarity between two images. By using these scores on every set of images, it is possible to construct the distance matrix between them.

The MSE matrix is divided in the 4 different damages states and the 100 repetitions of each measure. This means that the 100×100 first pixels give the MSE between the 100 repetitions of the healthy plate measurement in Figure 13. In these matrices, the images corresponding to the same damaged state are remarkably similar because their distances are small. In fact, all 100×100 squares following the diagonal have the lowest values of MSE. On the other hand, the distance between images from different damage state are bigger, that means that it is possible to separate the images and then the damage states. The persistence images generated with the magnet damage is more similar to the healthy plate than other ones. This could be an issue for the classification step.

Impact of the slicing window size

The slicing window size is a crucial parameter for classification efficiency. Because of the fixed shape of the time series of 900 μs, which corresponds to 900 points, the window size is selected to respect the integer factorization. It is also necessary to avoid partial slicing at the end of the signal. For this reason, the values of the parametric study are presented in Table 4:

For every configuration in Table 4, the MSE is computed on each window to compare the persistence images. The minimum window size is set to 9 μs because it is needed to generate the persistence diagrams with enough points in the time series. In fact, the narrower the slicing window is, the less significant the topological features are. By doing this, the results can be stored inside a similarity matrix as in Figure 13. This way, it is possible to estimate the ability of the LSF to regroup the same damage data and to separate damages with each other (Figure 14).

Because each computation is done here for 10 repetitions, the similarity matrix is (40×40). To reduce the matrix dimension, a pooling operation[28–30] is computed to extract the worst and the best cases. This gives two matrices of shape (4×4), one from the max pooling ($M_{\max}$) and one for the min pooling ($M_{\min}$). Two similarity indexes are proposed to compare the similarity and the diversity of the sample by changing the window size. The first tool is the simple matching coefficient (SMC):

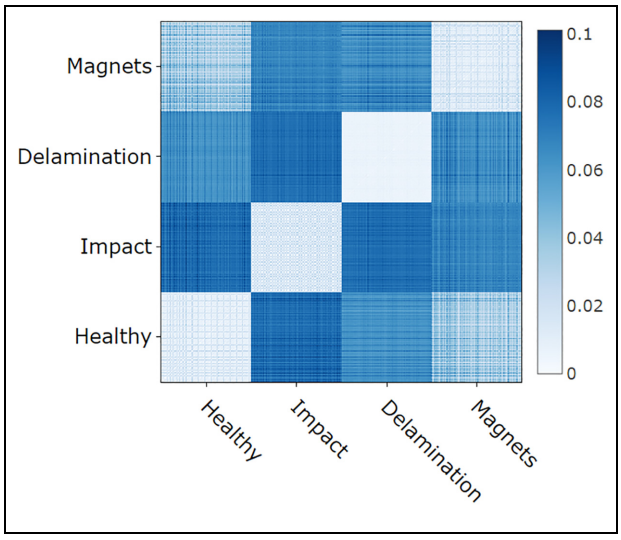

Figure 13. MSE score between each damage states images on the path [2–1].
MSE: mean square error.

$$SMC(M_{n,n}) = \frac{\sum_{i=0}^{n} m_{ii}}{\sum_{i=0}^{n}\sum_{j=0}^{n} m_{ij}} \tag{7}$$

With these definitions, the worst similarity matrix are multiples of the identity matrix with a $\mathrm{SMC}(k \cdot I_{n,n}) = 1$ and the best configurations are the matrices $\mathrm{BM}_{n,n} = k \times \left([1]_{n,n} - I_{n,n}\right)$ when $\mathrm{SMC}(\mathrm{BM}_{n,n}) = 0$ and for $k \in \mathbb{R}$.

To optimize the classification learning, the goal is to maximize the SMC index. Because each computation is done for each different paths and for 10 repetitions, the worst case will always be used. The worst case is computed using the max pooling on the diagonal values and the min pooling on the other. The worst similarity matrix is computed as follows:

$$\mathrm{WM}_{n,n} = \left[\mathrm{pooling}_{\max}(M_{n,n}) \cdot \delta_{i,j} + \mathrm{pooling}_{\min}(M_{n,n}) \cdot \left(1 - \delta_{i,j}\right)\right]_{i,j \in n}$$
$$\text{with } \delta_{i,j} \text{ the Kronecker delta}$$

This classification score is computed for each path on every slicing configuration and the results are presented in the following figure:

Figure 15 shows the estimation of the effectiveness of the classification for each slicing configuration. To maximize this efficiency of the classification model, the similarity indexes must be minimized. First, the maximum value on each path of the SMC is decreasing with the reduction of the window size. Decreasing the window size below 25 μs deteriorates the SMC score. On the other hand, wide windows will decrease the classification abilities because SMC indexes rise (Figure 16).

This parametric study gives an optimized range from 25 to 75 µs. In addition, the smaller the window is, the higher is the number of persistence images. More persistence images, mean more computation cost to generate the images, to process the data and to fit the classification model. In this article, the time duration of the input signal $\left(3 \cdot 10^{-5}\ \mathrm{s}\right)$ is chosen to slice the time series. As shown in the previous section 30 µs is inside the optimal range for the slicing window size. Moreover, parameter to consider should be the stride of the slicing windows to add recovering information.

Table 4. Window size values and corresponding number of windows.

Window size (µs)	9	18	25	30	36	50	75	90	150	180	450
Number of windows	100	50	36	30	25	18	12	10	6	5	2

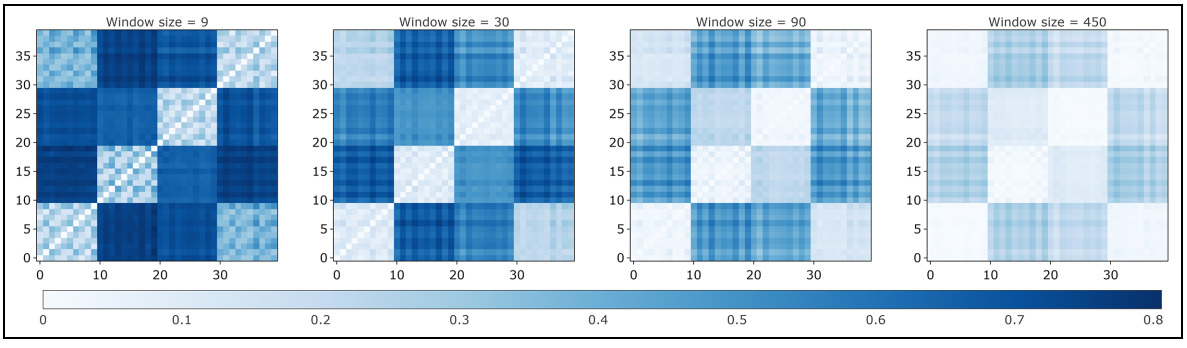

Figure 14. Similarity matrix computed with MSE on the path [3–5].

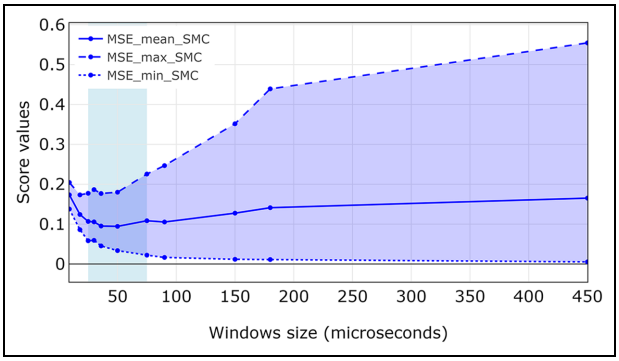

Figure 15. SMC classification score by the size of the slicing windows.
SMC: simple matching coefficient.

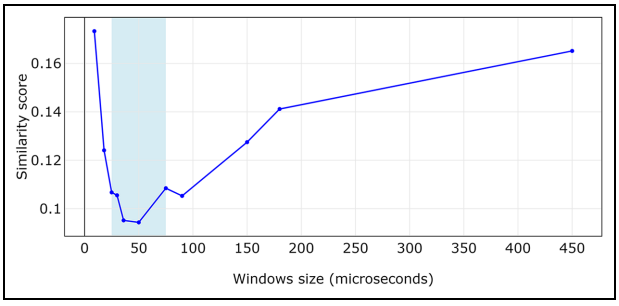

Figure 16. SMC index average value on each wave path.
SMC: simple matching coefficient.

By doing so, the number of time series to manipulate increases which implies more persistence diagrams. This parameter will not be studied in this article, and the stride value will be equal to the window size to avoid any recovering. It could also be interesting to use a better rephasing algorithm to fit the slices with the exact waves time of arrival on the sensor.

Classification using CNNs

The first tool which enables a comparison between two diagrams is the Wasserstein and Bottleneck distance matrix. As explained in the Definition 2, these matrices give a distance between several diagrams thanks to *transportation theory* and Euclidian distances. With this, it should be possible to separate the healthy case from damaged one. By doing so, it is possible to feed a Machine Learning algorithm with this distance in order to compute classification or regression. As the *persistence diagram* strongly depends on the time series shapes, most of the time, two diagrams corresponding to two different time series will not have the same number of points. That fact is an issue because Wasserstein distance needs to have the same shapes for the two input diagrams. The issue can be avoided by applying a persistence filter to delete the less persistent points in the diagrams, but this implies an important information loss. For example, if the two diagrams to compare contain 3 points for the first and 10 for the second, it will be necessary to suppress the 7 less persistent points in the second diagram which could be a loss of information. An alternative solution is to use persistence image to make the classification. Using the persistence images rather than persistence diagrams will not change the topological features processed because they wrap the same information but in two different spaces. This type of classification is based on Machine Learning methods for images. CNNs are very popular for image classification[31–33] because they produce better results than other Machine Learning methods with complex images. It resolves the input dimension problem because it admits the whole image when fully connected neural network needs 1D input data. The neural network used to achieve the classification is shown in Figure 17, and each neural layer is detailed in Table 5. Using CNN induces to modify the shape and the number of images through the convolution and pooling layers. $A\times(B;C)$ means that the format of the data is A images of shape (B; C). For the next applications, only the manipulation corresponding to one path will be considered. All the approach will be repeated for each different path in order to give a result.

The optimizer used to compile the CNN model is ADAM[34] with a learning rate equal to 10^{-4} and the loss is cross-entropy. The cross-entropy loss is computed following this equation:

$$H(p,q) = -\sum_{x\in X} p(x)\log(q(x)) \tag{8}$$

with p the predicted value and q the true value of the label and represented the loss between the predicted and the real output in order to train the network. To ensure the reliability of the network results, a cross-validation has been performed. This means that several loops of training and testing have been processed when the training and test dataset were randomly picked in the entire dataset. During the training, the model is learning with 15 epochs, this is enough to get an accuracy of 1.0 on most validation. The dataset is split into three sets: 64% for the training set, 16% on validation and the last 20% is the test batch. The training, validation and test sets are picked up by taking care that the amount of data corresponding to each damage are close. In fact, the neural network needs to train on balanced input data to improve classifying all damages. If one class of data is overrepresented, the CNN will be influenced by biases and could prevent to classify other damages. Most of the time accuracy converge to the maximal value on the test set. Figure 18 shows how

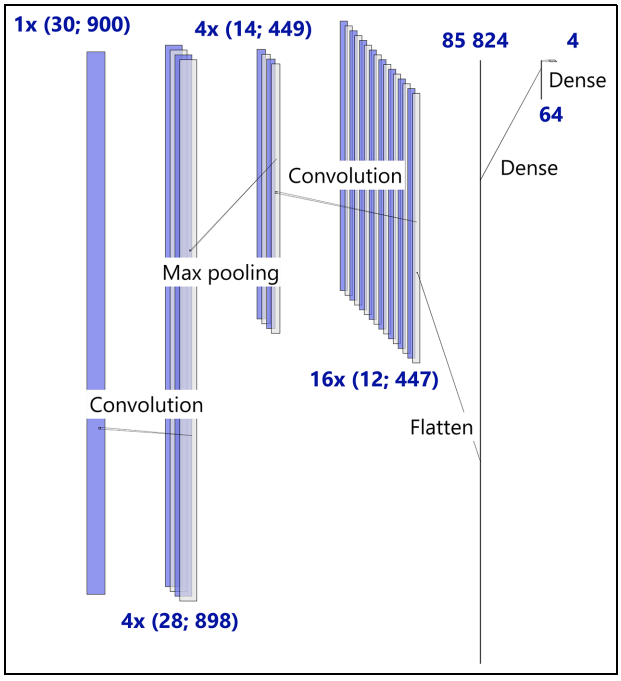

Figure 17. CNN for classification.
CNN: convolutional neural network.

this neural network is learning to classify the dataset. The training step is done using a cross-validation technics. This means that the initial dataset is divided into 5 training batch. Each batch will be tested as testing set when the 4 others will operate the fitting step. For this reason, 5 results are given and displayed in Figure 18. Each configuration is numbered from 0 to 4 and the CNN reach 100% accuracy on 4 of them.

After the training step, the neural network has to predict test data in order to verify if it has learnt well to analyse new unknown persistence images. Figure 18 displays that the neural network has learned well, with a 100% of accuracy, for classes of impacts and delamination damages. The accuracy corresponds to the ratio between number correct prediction and the number of total predictions done by the network, that is:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

where TP is the true positive, TN is the true negative, FP is the false positive and FN is the false negative.

However, there is still some confusion between the healthy plate and the damage generated by the magnet. These results were exposed before on the image comparison section. In fact, the persistence images from healthy and magnet damage are more similar than the two other classes. The results in Figure 19 show that the neural network is well learning to classify the damages for each training batches. Each score in the confusion matrix corresponds to one prediction for a single time series by the neural network. The test set is 20% of the 400 time series dataset, which corresponds to 80 time series.

The classification process is trained on each path signal. That makes the learning network dependent to the position of the actuator and the sensor. The same training phase has been computed for every path, and the results have been concatenated in a single confusion matrix in Figure 20.

Figure 20 indicates that the classification error between the healthy plate and the magnet damage is occurring but is not very common. This could be explained by the amount of energy generated with magnets perturbation. In fact, impacts and delamination return more energy and that make them easier to detect and classify. This effect is the same occurring on the MSE and structural similarity index calculation in the previous persistence image comparison section. The method presented in this section must be applied on each path. This means that the algorithms will return a result for each different path stored in the dataset. The last step to compute a unique decision about the health of the structure is to gather all the results in a scalar,

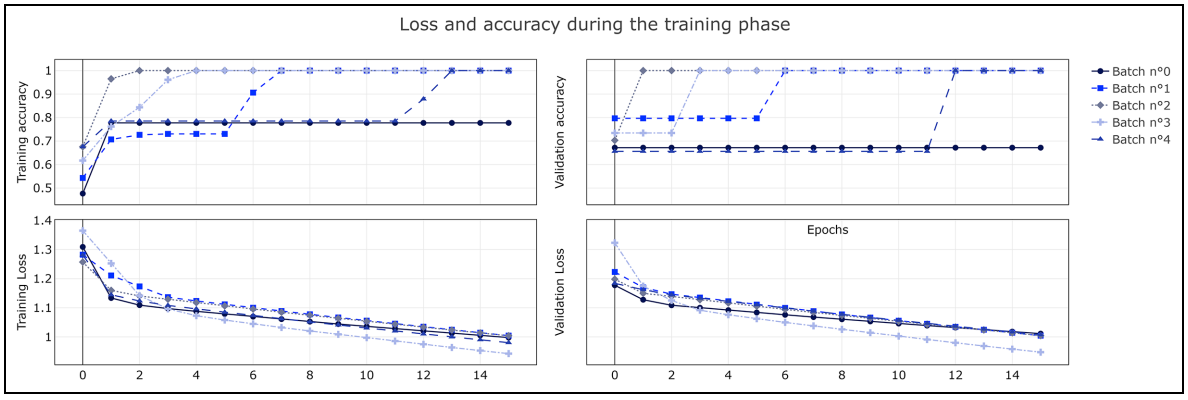

Figure 18. Loss and accuracy evolution during training for several training batches.

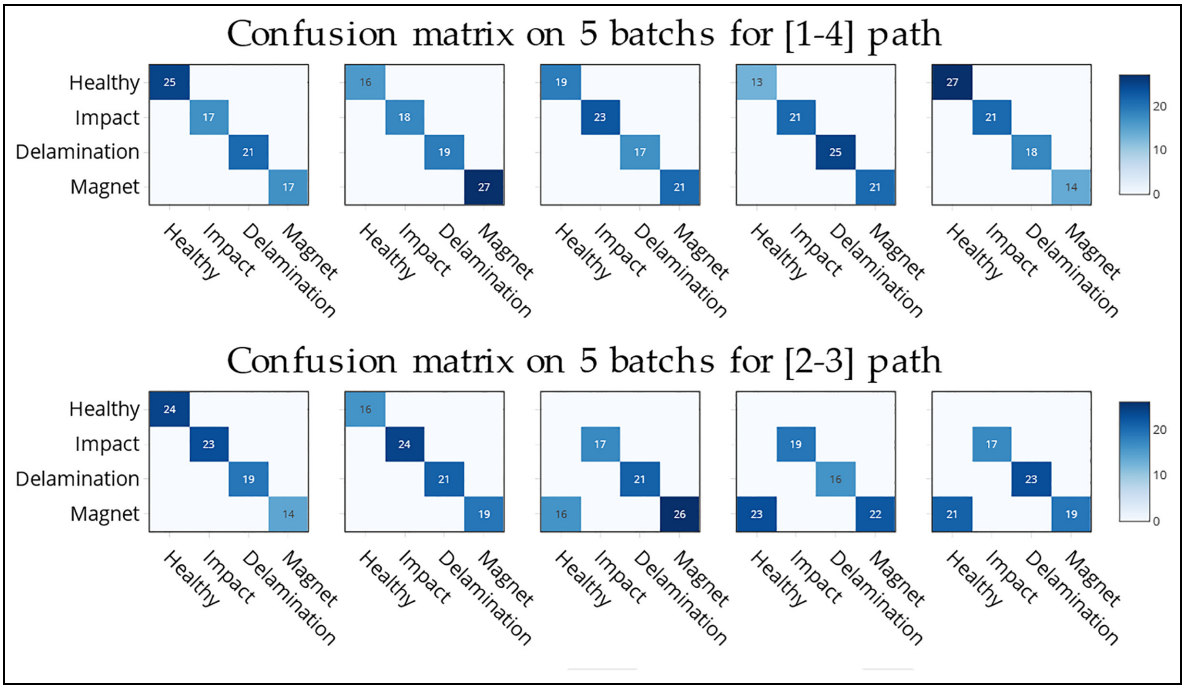

Figure 19. Confusion matrix for five test batches.

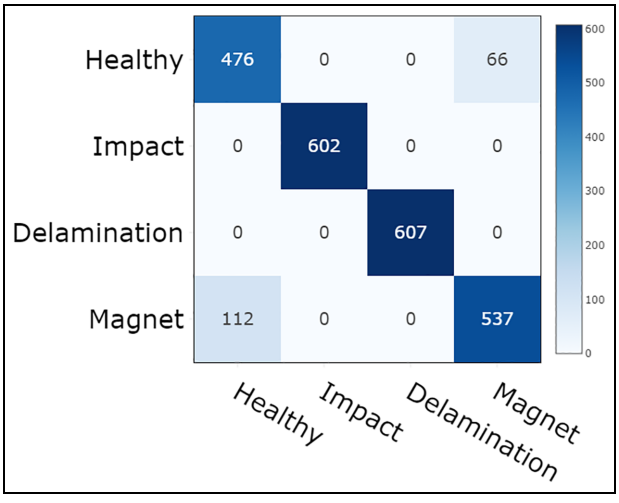

Figure 20. Test confusion matrix sum on every paths.

for example the mean function could be a solution or a voting-based consensus algorithm.[35,36]

Conclusion and perspectives

This article presents a damage detection and classification method by processing the wave propagating on structural part. This approach provides an original solution to SHM research field. Many data processing works are now interested in TDA tools to solve detection and classification problems on multi-dimensional data. These tools open a new paradigm to solve SHM issues because it does not analyse information on a strict physical wave-based way. One of the main difference between current SHM works[2,5,37] and the TDA application method is that most of the time frequency analysis are based on the analysis of the difference of

Table 5. Neural network architecture and parameters.

No.	Layer type	Parameters	Shapes output	Number of learning weights
1	Convolutional layer	Kernel: (3,3) Activation: Relu	$4\times(28,\ 898)$	40
2	Max pooling	Pool size: (2,2)	$4\times(14,\ 449)$	0
3	Convolutional layer	Kernel: (3,3) Activation = Relu	$16\times(12,\ 447)$	592
4	Flatten	$\varnothing$	$1\times(85,824)$	0
5	Dense layer	Activation: Softmax	$1\times(64)$	5,492,800
6	Dense layer	Activation: Linear	$1\times(4)$	260

two signals when TDA tools use directly the measured guided waves. The article displays the interest of TDA to characterize and analyse the topological data coming from waves. In fact, the four damages type such as impacts, delamination, stiffness perturbations and healthy case can be predicted by analysing the topological features of the data coming from the piezo transducers. Applying TDA tools such as LSF on raw data without any physical understanding is not a good way to solve our problem. But better results can be found by slicing the temporal data into several windows. In fact, each windows generate unique persistence image with the LSF method and enable to extract convenient topological feature on each data. By this way, it is possible to show the difference and the similarities of the features between each damage type and then to classify them. A simple neural network with convolutional layers can classify the images with few images and few layers. That makes the training step very convenient and fast and could be transferred on many more experimental and simulated data. This work also shows that the topological analysis parameters are very important in order to separate the persistence homology of each damage. The damages are not on the same location because they were manufactured to test several approaches for several goals like detection, classification, localization and quantification steps of SHM process. In fact, it is possible to see the problem in a different way by considering that a damage is defined by a damage type and a localization. Testing this method on a single damage for different locations or on a single location for several damage could be interesting to show the ability to classify the damage type alone or to localize the damage.

This approach could be used for future works to estimate the location or the size of the damage inside the structure (learning a regression instead of the classification presented here). TDA technics provide many ways to analyse the data, the slicing window method[23] is another approach to analyse time series which have a cyclic behaviour. The TDA opens a lot of methods and representations to make classification or regression. For example, it is possible to use the persistence diagrams by considering them as point clouds. The Betti number and other representation, which are not presented in this article, could be useful in future works such as persistence barcodes and landscapes.[18,38] Because of the experimental data analysis, the current results are linked to the physical properties of the plates and the PZT. Similar work on different materials, geometries and transducers could give a better vision about the global purpose of TDA for SHM application. Moreover, the methods developed in this article are analysing waves from one actuator measured by a single sensor. This mean that the neural network learns to classify the damages on a single wave path. Even if the classification results are good for each path, it is important to concatenate the different prediction of damage for each path to use all accessible information. This kind of gathered decision-making is an entire computer science field where many solutions provided by research exists. For example, the voting-based consensus algorithms could be a simple way to achieve this kind of operations. Transfer learning technics could provide a way to extract common topological features from time series. Another way to use TDA is to consider each time series as a spatial axis. This would provide a way to write the measured data as a multidimensional point cloud and enable to use Vietoris–Rips filtration. The TDA method for SHM classification could be an alternative or an improvement for other methods. In fact, it could be possible to use classical SHM method and add some TDA features to get better results. In any case, comparing the method with other one is interesting to be able to have a critical eye on it; this is why it will be done in future works. It is worth noting that as the efficiency of an SHM process is strongly dependent on structure and on the damage types, measuring the robustness of the TDA method to classify the damage on different structures and damages is a crucial step to validate the potential of this method on real structures. This is why testing methods on several structure shapes should be the best way to compare them. A benchmark comparison with other features generated by other methods will be carried out in future work.

Declaration of conflicting interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

ORCID iDs

Arthur Lejeune https://orcid.org/0000-0001-5904-548X
Marc Rébillat https://orcid.org/0000-0003-0469-8437

References

1. Fouladirad M, Grall A and Dieulle L. On the use of online detection for maintenance of gradually deteriorating systems. *Reliab Eng Syst Saf* 2008; 93: 1814–1820.
2. Briand W. *Lamb waves based active sparse tomography for damage size quantification in composite structures: data-driven and parameter inversion methods*. PhD Thesis, HESAM Université, 2022.

3. Rahbari A, Rébillat M, Mechbal N, et al. Unsupervised damage clustering in complex aeronautical composite structures monitored by Lamb waves: an inductive approach. *Eng Appl Artif Intell* 2021; 97: 104099.
4. Miller CA and Hinders MK. Multiclass feature selection using computational homology for Lamb wave-based damage characterization. *J Intell Mater Syst Struct* 2014; 25: 1511–1527.
5. Guo S, Rébillat M and Mechbal N. Prediction of frequency and spatially dependent attenuation of guided waves propagating in mounted and unmounted A380 parts made up of anisotropic viscoelastic composite laminates. *Struct Health Monit* 2023; 22: 1326–1352.
6. Tibaduiza-Burgos DA and Torres-Arredondo MA. Investigation of an expert health monitoring system for aeronautical structures based on pattern recognition and acousto-ultrasonics. *Smart Mater Struct* 2015; 24: 085020.
7. Pavlopoulou S, Worden K and Soutis C. Novelty detection and dimension reduction via guided ultrasonic waves: damage monitoring of scarf repairs in composite laminates. *J Intell Mater Syst Struct* 2016; 27: 549–566.
8. Borate P, Wang G and Wang Y. Data-driven structural health monitoring approach using guided Lamb wave responses. *J Aerosp Eng* 2020; 33: 04020033.
9. da Silva S, Villani LGG, Rébillat M, et al. Gaussian process NARX model for damage detection in composite aircraft structures. *J Nondestruct Eval Diagn Progn Eng Syst* 2021; 5: 011007.
10. Gidea M. Topological data analysis of critical transitions in financial networks. In: *Springer proceedings in complexity*. Tel-Aviv, Israel: Springer, 2017, pp. 47–59.
11. Iniesta R, Carr E, Carrière M, et al. Topological data analysis and its usefulness for precision medicine studies. *SORT – Stat Oper Res Trans* 2022; 46: 115–136.
12. Rabadán R and Blumberg AJ. *Topological data analysis for genomics and evolution: topology in biology*. Cambridge, UK: Cambridge University Press, 2019.
13. Skaf Y and Laubenbacher R. Topological data analysis in biomedicine: a review. *J Biomed Inform* 2022; 130: 104082.
14. Salch A, Regalski A, Abdallah H, et al. From mathematics to medicine: a practical primer on topological data analysis (TDA) and the development of related analytic tools for the functional discovery of latent structure in fMRI data. *PLoS One* 2021; 16: e0255859.
15. Seversky LM, Davis S and Berger M. On time-series topological data analysis: new data and opportunities. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, 2016, pp. 59–67.
16. Gowdridge T. On topological data analysis for SHM: an introduction to persistent homology. In: *Data science in engineering proceedings of the 39th IMAC, a conference and exposition on structural dynamics*, 2021, pp. 169–184.
17. Bauer U. Ripser: efficient computation of Vietoris-Rips persistence barcodes. *J Appl Comput Topol* 2021; 5: 391–423.
18. Ghrist R. Barcodes: the persistent topology of data. *Bull Am Math Soc* 2007; 45: 61–76.
19. Flamary R, Cuturi M, Courty N, et al. Wasserstein discriminant analysis. *Mach Learn* 2018; 107: 1923–1945.
20. Bauer U, Kerber M, Reininghaus J, et al. Phat – persistent homology algorithms toolbox. *J Symb Comput* 2017; 78: 76–90.
21. Cawley P. Closing the gap between research and industrial deployment. *Struct Health Monit* 2018; 17: 1031–1330.
22. Lizé E, Rébillat M, Mechbal N, et al. Optimal dual-PZT sizing and network design for baseline-free SHM of complex anisotropic composite structures. *Smart Mater Struct* 2018; 27: 115018.
23. Perea JA and Harer J. Sliding windows and persistence: an application of topological methods to signal analysis. *Found Comput Math* 2015; 15: 799–838.
24. Tauzin G, Lupo U, Tunstall L, et al. giotto-tda: a topological data analysis toolkit for machine learning and data exploration. *arXiv:2004.02551*.
25. Tralie C. Lower Star Time Series – Ripser.py 0.6.4 documentation, https://ripser.scikit-tda.org/en/latest/notebooks/Lower%20Star%20Time%20Series.html (2019, accessed 19 January 2023).
26. Boissonnat J-D, Dey TK and Maria C. The compressed annotation matrix: an efficient data structure for computing persistent cohomology. In: Bodlaender HL and Italiano GF (eds) *Algorithms – ESA 2013*. Berlin, Heidelberg: Springer, 2013, pp. 695–706.
27. Edelsbrunner H and Harer J. *Computational topology: an introduction*. Providence, RI: American Mathematical Society, 2010.
28. Sun M, Song Z, Jiang X, et al. Learning pooling for convolutional neural network. *Neurocomputing* 2017; 224: 96–104.
29. Data Science and Machine Learning. What is pooling in deep learning?, https://www.kaggle.com/discussions/questions-and-answers/59502 (2018, accessed 11 June 2023).
30. Gholamalinezhad H and Khosravi H. Pooling methods in deep neural networks: a review. *arXiv:2009.07485*.
31. Sharma N, Jain V and Mishra A. An analysis of convolutional neural networks for image classification. *Procedia Comput Sci* 2018; 132: 377–384.
32. Lecun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition. *Proc IEEE* 1998; 86: 2278–2324.
33. Guo T, Dong J, Li H, et al. Simple convolutional neural network on image classification. In: *2017 IEEE 2nd international conference on big data analysis (ICBDA)*, Beijing, China, pp. 721–724. IEEE.
34. Habib G and Qureshi S. Optimization and acceleration of convolutional neural networks: a survey. *J King Saud Univ Comput Inf Sci* 2022; 34: 4244–4268.
35. Li K, Li H, Wang H, et al. PoV: an efficient voting-based consensus algorithm for consortium blockchains. *Front Blockchain*; 3: 1–15.
36. Ayad HG and Kamel MS. On voting-based consensus of cluster ensembles. *Pattern Recognit* 2010; 43: 1943–1953.
37. Fendzi C, Mechbal N, Rébillat M, et al. A general Bayesian framework for ellipse-based and hyperbola-based damage localization in anisotropic composite plates. *J Intell Mater Syst Struct* 2016; 27: 350–374.

38. Gidea M and Katz Y. Topological data analysis of financial time series: landscapes of crashes. *Physica A* 2018; 491: 820–834.
39. Vietoris L. Über den höheren Zusammenhang kompakter Räume und eine Klasse von zusammenhangstreuen Abbildungen. *Math Ann* 1927; 97: 454–472.
40. Zhang Y. Persistent homology and sparse Vietoris-Rips filtration. The University of Chicago Mathematics REU, http://math.uchicago.edu/~may/REU2017/REUPapers/Zhang,yujie.pdf (2017, accessed 12 January 2023).
41. Adams H, Emerson T, Kirby M, et al. Persistence images: a stable vector representation of persistent homology. 35.
42. Getreuer P. A survey of Gaussian convolution algorithms. *Image Process Line* 2013; 3: 286–310.

Appendix: TDA and persistent homology

For clarity's sake, we will only consider two-dimensional dataset to apply TDA tools, but it is important to remember that the theory is useful for every dimension data. We will focus on the most used algorithm for TDA named *Vietoris–Rips filtration*, but it is not the only method to construct a *simplicial complex* from a dataset. Čech complex and Delaunay complex are also used in special situations.[27] The two most significant topological objects used in TDA are the following ones.

A k-simplex is the equivalent element for the triangle in k dimension. To construct a k-simplex, $k + 1$ vertices are necessary. A k-simplex is like a complete graph of $k + 1$ nodes. This means that a 0-simplex is a point, a 1-simplex is a segment and a 2-simplex is a triangle. A simplex can be described as a set of lower dimension objects. These objects are called n-face with n giving the dimension. The 1-faces of a 2-simplex are the edges of the triangle when 0-faces are these vertices (Figure A1).

A simplicial complex is a set of simplices that must respect two rules. Let be K a simplicial complex:

- Every face of simplices of K is contained in K.
- The intersection of two simplices in K is empty or a n-face of both

Vietoris–Rips filtration

Today, the Vietoris–Rips filtration is one of the most important algorithms used in TDA. It uses the Vietoris–Rips complex, which appears in Vietoris work[39] and later in Rips work.

Let (X, ∂_X) be a finite metric space with ε the filtration parameter. $\partial_X(x_i, x_j)$ defines the distance between the two points x_i, x_j. The Vietoris–Rips complex contains the k-simplex σ_k according to the following condition:

$$\sigma_k = \{x_0, \ldots, x_k\} \in \mathrm{VR}_\varepsilon \quad if \ \partial_X(x_i, x_j) \leqslant 2\varepsilon, \quad i,j \leqslant k$$

ε must be positive. By increasing ε step by step, the algorithm constructs a set of Vietoris–Rips complexes. At $\varepsilon = 0, \mathrm{VR}_\varepsilon = \varnothing$ and $\varepsilon = \max(\partial_X(X)), \ \mathrm{VR}_\varepsilon = X$.

The Vietoris–Rips complex has a graphic representation. In two-dimensional space, ε is analogue to the radius of circles (and sphere in 3D) drawn around each point of X (Figure A2).

The Vietoris–Rips complex constructed is an *abstract simplicial complex*. For each value of ε, the complex defines a connection between points. By joining complexes for several values of ε, it is possible to construct a filtration. In fact, if $0 \leqslant \varepsilon_1 < \varepsilon_2, \ \mathrm{VR}_{\varepsilon_1}(X) \subseteq \mathrm{VR}_{\varepsilon_2}(X)$.

Homology definition

The homology of a topological space a set of invariant features of X, it is represented as homology groups written $H_k(X)$. k corresponds to the dimension of features, and X is the topological space which is in practice the dataset. H_k gives information about k-dimensional topological features (also called holes) inside the data.

To find the *homology groups*, it is necessary to define the *boundary mapping* of a k-chain or a k-simplex. The *boundary* of a p-chain is the sum of the boundaries of its simplices. With the definition, it is possible to make a mapping using the boundary function (Figure A3).

The *homology group*, $H_p(K) = Z_p(K)/B_p(K)$. In other words, pth homology group consists of p-cycles that are not boundaries of the C_{p+1} group.

With geometrical meanings, being a p-cycle means that the object forms the envelope of a p-dimensional space. Meanwhile, not being p-boundary means that this p-dimensional interior does not belong to the *underlying space*. This supports the idea that homology express information about the p-dimensional holes. 1D holes are described by $H_1(K)$ and can be seen as circular holes. 2D holes are cavities, etc. Nevertheless, $H_0(K)$ makes an exception because it encodes information about connected components in the simplicial complex K.

The *Betti number* is a topological invariant, and it represents the number of p-dimensional homology class in the simplicial complex K. It is denoted as $\beta_p = \mathrm{rank}(H_p)$.

Persistence homology

It is possible to extract *homology classes* for each homology dimension from H_0 to H_p for a p-dimensional simplicial complex. Considering *homology groups and classes* at a single state will not provide a lot of information about the topological features of data. That is why it is necessary to use what is called persistence homology. To do so, it is needed to consider the sequence of *chain complexes* among the filtrations.

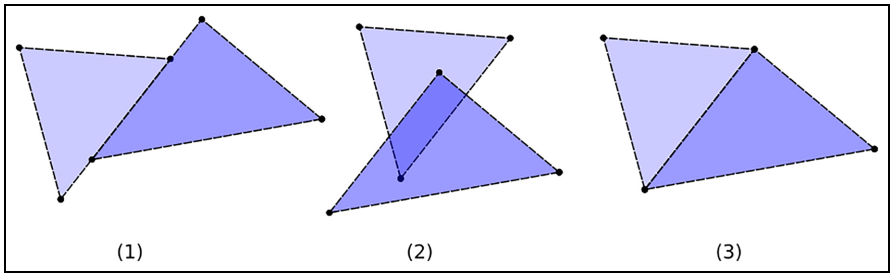

Figure A1. Simplicial complex computation. Only (3) is a valid Simplicial complex.

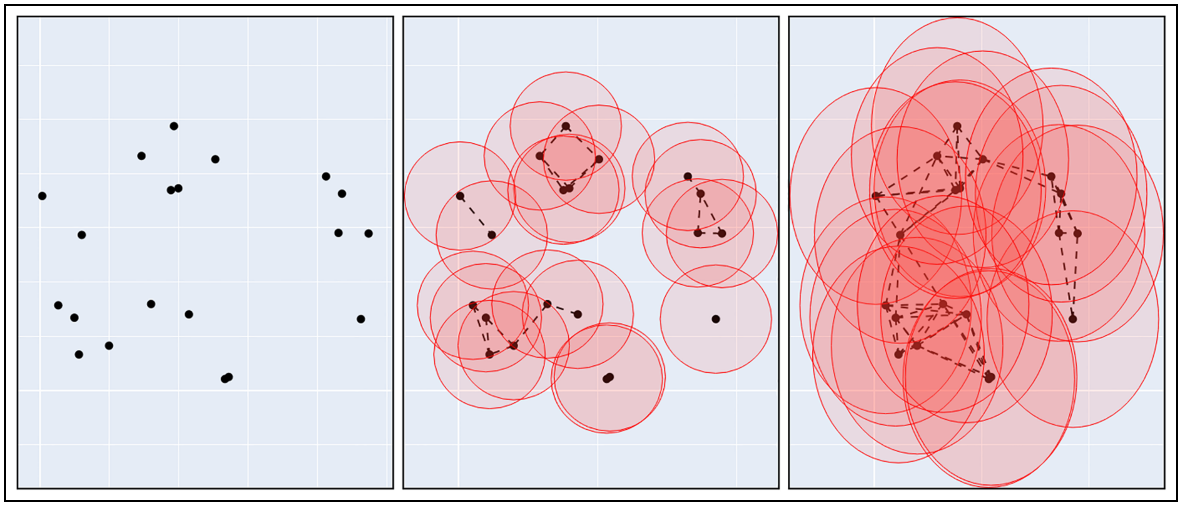

Figure A2. The Vietoris–Rips complex from a data point cloud.

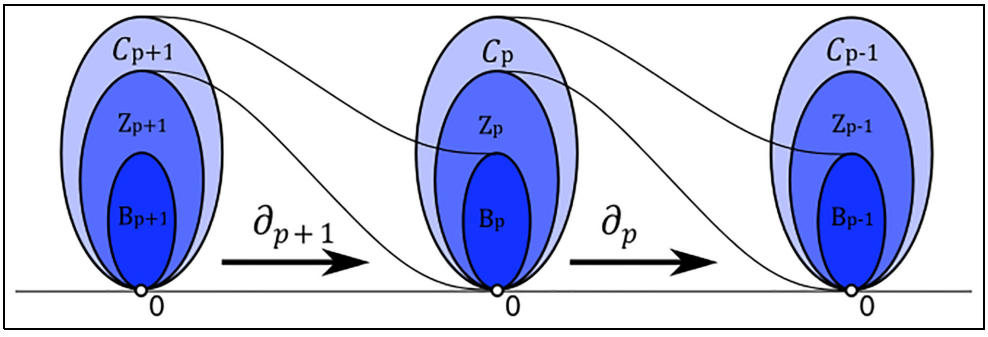

Figure A3. Boundary mapping.

$F_p^j : C_p^j \to C_p^{j+1}$ with $C_p^j = C_p(X^j)$ is the *filtration mapping*, this is the direct mapping of the filtration inclusion. The composition of several homomorphisms is denoted $F_p^{i,j} : C_p(X^i) \to C_p(X^j)$.

$\partial_p^j : C_p^j \to C_{p+1}^j$ is the *boundary mapping*.

The two-dimensional diagram shows the mapping for each filtration by the *boundary map*.[40] This representation will be helpful to understand the homologies persistence:

The *persistence homology* consists of catching the moment; the precise value of the filtration parameter ε when a *homology class* appears and disappears. These moments are called *births and deaths*.

Let $i,j \in \mathbb{N}$ $i<j$,

Figure A4 shows the [c] homology class be born at index i and die at index j.

Following the filtration index, births and deaths can be considered as index or equal to the value of the

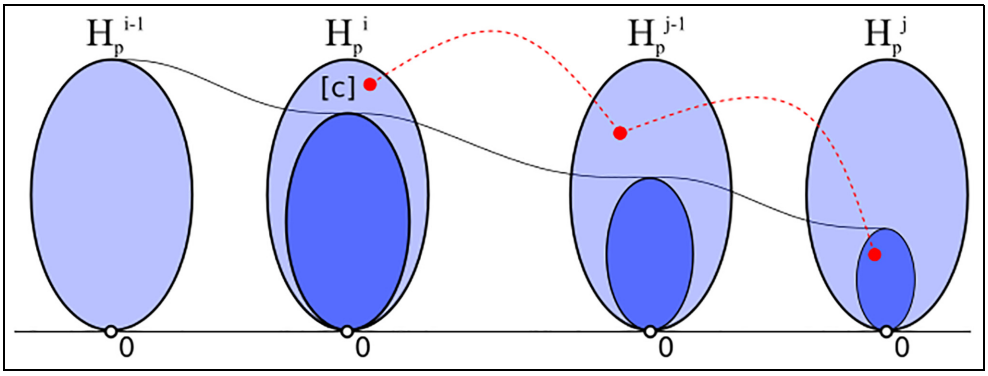

Figure A4. Birth and death of [c].

filtration parameter. Both are related with a bijective morphism, so this does not change the result.

The difference Death − Birth is called *persistence*. *Persistence* is always positive as Death ⩾ Birth. Sometime, *persistence* is also denoted *lifetime* because it refers to the delta between birth and death.

Persistence representation. Once the simplicial complexes have been constructed, invariants or homology groups for dataset can be identified. Then, we can use the concept of persistent homology to encode the evolution of the homology (or topology) of the classes of nested spaces. The goal is to adapt the algorithm to keep track of a homology basis and pairs positive simplices (birth of a new homological class) to negative simplices (death of an existing homology class).

Persistence diagrams. The persistence diagram is a two-dimensional scatter chart with homology class *birth* as x-axis and *deaths* as y-axis. The birth value is always inferior to death, this implies that all points are situated above the $y = x$ function. All homology classes are represented on the same graph. Dealing with two-dimensional data implies to represented homology of dimensions 0 and 1.

Figure A5 shows the homology persistence of the circle point cloud. Every time, H_0 homology is born at 0 index, that is why H_0 point is on the $y = 0$ axis. H_0 homology deaths correspond to the geometric pairwise distance between point in the dataset. The most important information that can be extracted is the H_1 class represented by the left and top points in the diagram. This means that only one topological one-dimensional hole is contained in the dataset. This result is right comparing to the topology theory. In fact, an ellipse in a two-dimensional space is considered as a one-dimensional homology class.

Persistence images. The persistence images are computed by discretization on the persistence diagram space. This representation allows to manipulate the same shape image rather than different shape diagrams. The discrete space image is not operable directly because it does not describe well several points close to each other. To improve the representation, a gaussian kernel convolution[41,42] is always used. This convolution is based on a Gaussian distribution defined by its mean (u_x, u_y) and its variance (σ^2) in the formula:

$$g_u(x,y) = \frac{1}{2\pi\sigma}\exp\left(-\frac{(x-u_x)^2 + (y-u_y)^2}{2\sigma^2}\right) \quad \text{(A1)}$$

with $(x,\ y)$ the position of the homology in the persistence-death space.

Figure A6 shows a simple example of the discretization (b) and the effect of the gaussian convolution (c). The larger σ is, the larger will be the spot for each homology in the image.

To generate persistence image, it is necessary to split the persistence diagram into each homology dimension. Therefore, there are two different persistence images for two-dimensional data. In fact, the first image corresponds to H_0 homology group, and the second one is H_2 persistence image (Figure A7).

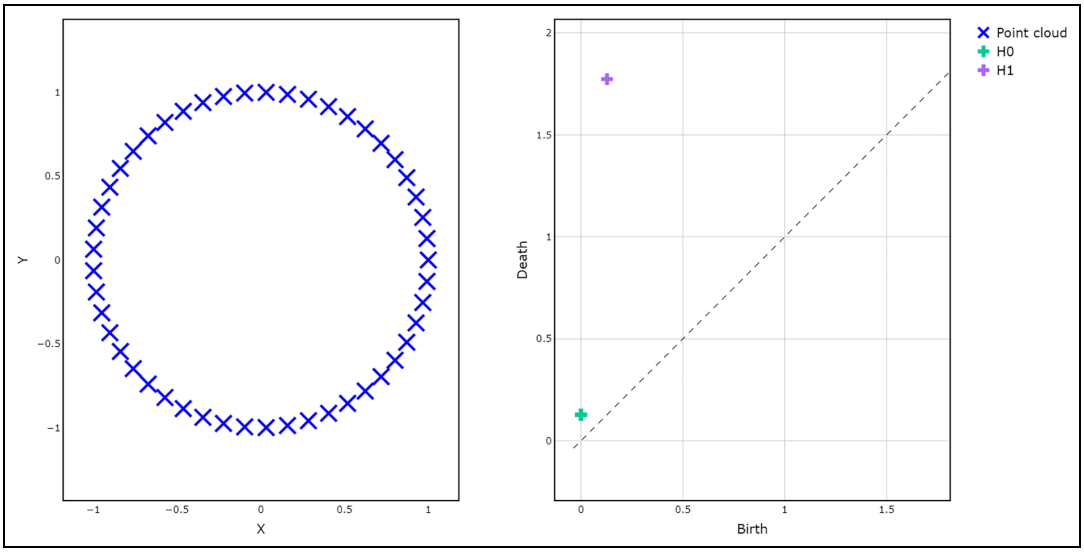

Figure A5. Persistence diagram computed with the Vietoris–Rips filtration on perfect circle point cloud (two dimensional).

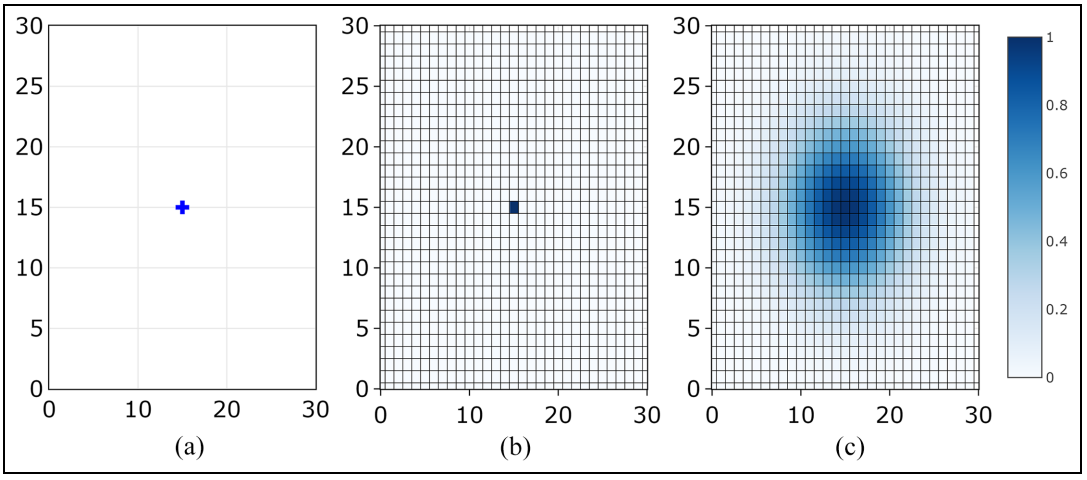

Figure A6. (a) Persistence diagram. (b) Persistence image before convolution. (c) Persistence image after convolution.

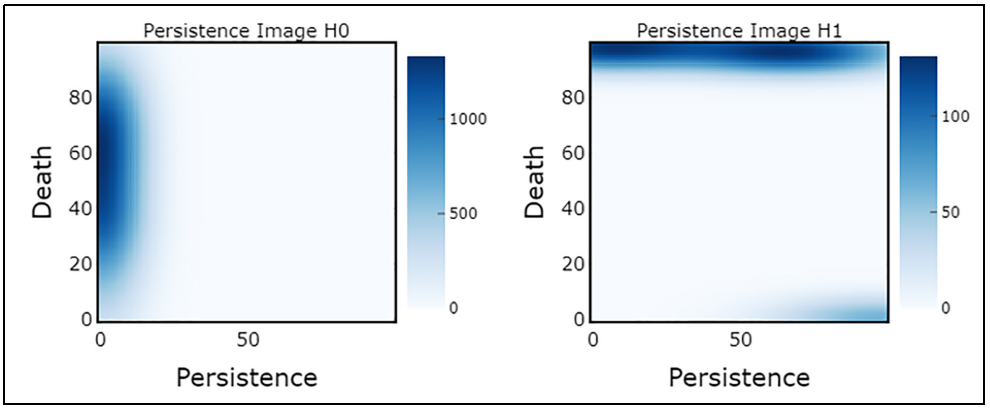

Figure A7. Persistence images from the circle point cloud.
Right: Persistence image of H_0 homology persistence. Left: Persistence image of H_1 homology persistence.