

Article

# Vibration-Based Structural Health Monitoring Using Piezoelectric Transducers and Parametric $t$ -SNE

David Agis  and Francesc Pozo \* 

Control, Modeling, Identification, and Applications (CoDALab), Department of Mathematics, Escola d'Enginyeria de Barcelona Est (EEBE), Universitat Politècnica de Catalunya (UPC), Campus Diagonal-Besòs (CDB), Eduard Maristany, 16, 08019 Barcelona, Spain; david.agis@upc.edu

\* Correspondence: francesc.pozo@upc.edu; Tel.: +34-934-137-316

Received: 12 February 2020; Accepted: 17 March 2020; Published: 19 March 2020



**Abstract:** In this paper, we evaluate the performance of the so-called parametric  $t$ -distributed stochastic neighbor embedding ( $P$ - $t$ -SNE), comparing it to the performance of the  $t$ -SNE, the non-parametric version. The methodology used in this study is introduced for the detection and classification of structural changes in the field of structural health monitoring. This method is based on the combination of principal component analysis (PCA) and  $P$ - $t$ -SNE, and it is applied to an experimental case study of an aluminum plate with four piezoelectric transducers. The basic steps of the detection and classification process are: (i) the raw data are scaled using mean-centered group scaling and then PCA is applied to reduce its dimensionality; (ii)  $P$ - $t$ -SNE is applied to represent the scaled and reduced data as 2-dimensional points, defining a cluster for each structural state; and (iii) the current structure to be diagnosed is associated with a cluster employing two strategies: (a) majority voting; and (b) the sum of the inverse distances. The results in the frequency domain manifest the strong performance of  $P$ - $t$ -SNE, which is comparable to the performance of  $t$ -SNE but outperforms  $t$ -SNE in terms of computational cost and runtime. When the method is based on  $P$ - $t$ -SNE, the overall accuracy fluctuates between 99.5% and 99.75%.

**Keywords:** classification; detection; parametric  $t$ -distributed stochastic neighbor embedding ( $P$ - $t$ -SNE); piezoelectric transducers (PZTs); principal component analysis (PCA); structural health monitoring (SHM); vibration-based SHM

## 1. Introduction

In structural health monitoring (SHM), an important process for engineering structures, many methods have been applied for damage detection. In [1], the use of the Treed Gaussian Process model—a class of powerful switching response surface model—is illustrated in the context of the SHM of bridges. In [2], a SHM methodology, based on the system-identification techniques, is proposed to quantify the structural degradation in laminated composite booms used in satellite application. In [3], it is casted SHM in the context of statistical pattern recognition, and damage or structural changes are detected using two techniques based on time series analysis. In [4], three optimization-algorithm based support vector machines for damage detection in SHM are presented, which are expected to help engineers to process high-dimensional data.

Real-world datasets usually have high dimensionality, and their dimensionality may need to be reduced to facilitate data processing. Dimensionality reduction is the process of reducing the number of high-dimensional variables by obtaining a low-dimensional set of variables. This reduced representation must correspond to the intrinsic information of the data. Dimensionality reduction is very important, because it alleviates undesired properties of high-dimensional spaces, such as “the curse of dimensionality” [5]. In the literature, various dimensionality reduction methods have been

proposed: (i) linear methods, such as principal component analysis (PCA) [6,7] and linear discriminant analysis (LDA) [8,9], and (ii) nonlinear methods, such as isometric mapping (ISOMAP) [10,11] and the non-parametric version of  $t$ -distributed stochastic neighbor embedding ( $t$ -SNE) [12].

In general, real-world data are likely to be highly nonlinear. Therefore, unsupervised nonlinear dimensionality reduction techniques are widely used in many applications for pattern recognition or classification [13,14], visualization [15,16], and compression [17] of big datasets. Among these types of techniques,  $t$ -SNE is extensively adopted. However, this method is not designed to support the incorporation of out-of-sample data—that is, the embedding of new data. To avoid this problem, the so-called parametric  $t$ -SNE (P- $t$ -SNE) method [18] was proposed. This method uses a neural network (NN) to learn an explicit parametric mapping function from a high-dimensional data space to a low-dimensional space. Thus, P- $t$ -SNE can incorporate out-of-sample data. Another advantage of P- $t$ -SNE is that it can be applied to large-scale datasets, while  $t$ -SNE can only be applied to datasets with a size not greater than the order of thousands.

To address the above-mentioned problems of  $t$ -SNE, in this work, we propose a strategy that combines PCA and P- $t$ -SNE to detect and classify damage to structures to diagnose. This combination is much better than a combination of PCA with  $t$ -SNE: the proposed method achieves similar embedding performance but with a much lower computational cost and runtime. This was confirmed by our experimental results on an aluminum plate instrumented with four piezoelectric transducers (PZTs). Therefore, the aims of this paper are (i) to compare two approaches (P- $t$ -SNE versus  $t$ -SNE) and (ii) to identify the advantages of the parametric version. To do this, we will use scenarios 1 and 3 from [19], because these two scenarios are two extreme cases. In addition, we will approach the damage detection and classification problem in the frequency domain, as recommended in the same paper [19]. The final classification of the current state of the structure is based on two different voting systems: the so-called majority voting and the sum of the inverse distances [19].

The contributions of this study are summarized in the following list. These contributions mainly refer to the data preprocessing, divided into three parts: data integration, data transformation, and data reduction.

1. Data preprocessing: Data integration. According to [20], there are six different ways to arrange the raw data collected by multiple sensors that collected measurements for several seconds in different experiments. The type of integration of raw data affects the posterior analysis. In this work, we have considered type- $E$  unfolding.
2. Data preprocessing: Data normalization. The second step, before data transformation, is the data normalization. We perform the mean-centered group scaling (MCGS), as detailed in [21].
3. Data preprocessing: Data transformation. We build the PCA model ( $\mathbf{P}$ ) so that the normalized data  $\mathbf{X}$  are transformed into the projected data  $\mathbf{T} = \mathbf{X}\mathbf{P}$ . Notably, matrices  $\mathbf{X}$  and  $\mathbf{T}$  have equal dimensions; hence, no reduction is performed at this stage.
4. Data preprocessing: Data reduction (phase I). We use PCA for the data reduction. The number of principal components  $\ell \in \mathbb{N}$  is chosen so that the proportion of the variance explained is greater than or equal to 95%.
5. Data preprocessing: Data reduction (phase II). We now propose P- $t$ -SNE, avoiding the limitations of  $t$ -SNE, as a second phase to reduce the dimensionality from  $\ell$  to 2. This is one of the first approaches that use P- $t$ -SNE in the field of SHM.
6. Data postprocessing: Classification. For the classification, we propose the majority voting and the sum of the inverse distances, where each actuation phase casts a vote.

As a summary, the contribution of the present work is, precisely, the combination of existing preprocessing methods with a very promising approach: P- $t$ -SNE.

The remainder of this paper is structured as follows. In Section 2, we present the P- $t$ -SNE method. Section 3 describes the preprocessing of the baseline data, reduction of the global dimension of the data, and creation of the clusters using P- $t$ -SNE. Section 4 describes the damage diagnosis procedure.

Subsequently, the application of the proposed method is presented in Section 5. Section 6 shows the results. Finally, Section 7 provides our conclusions.

## 2. Parametric $t$ -SNE (P- $t$ -SNE)

The non-parametric version of  $t$ -SNE has a huge computational cost of optimization: to map new data, the optimization has to run for the complete set again. To avoid the heavy optimization of the  $t$ -SNE, the P- $t$ -SNE was proposed. P- $t$ -SNE is an unsupervised dimensionality reduction technique that learns a parametric mapping between the high-dimensional data space and the low-dimensional latent space, preserving the local structure of the data in the latent space as well as possible.

In the P- $t$ -SNE method, the mapping  $f : X \rightarrow Y$  from the high-dimensional space  $X$  to the low-dimensional space  $Y$  is parameterized through a feed-forward NN with weights  $W$ . The NN is trained in such a way that it retains the local structure of the data in the low-dimensional space. There are two main stages in the training procedure:

1. Pretraining with a restricted Boltzmann machine (RBM). RBM is used to construct a pretrained P- $t$ -SNE network. The main aim of the pretraining stage is to define an initialization of the model parameters for the next stage.
2. Fine-tuning using the cost function of P- $t$ -SNE. In this stage, the weights of the pretrained NN are fine-tuned in such a way that the NN preserves the local structure of the data in the low-dimensional space. In the feed-forward NN, term  $q_{ij}$  [12,19] of the  $t$ -SNE is adapted as follows:

$$q_{ij} = \frac{[1 + \|f(x_i|W) - f(x_j|W)\|^2/\alpha]^{-(\alpha+1)/2}}{\sum_{k \neq i} [1 + \|f(x_k|W) - f(x_i|W)\|^2/\alpha]^{-(\alpha+1)/2}}$$

where  $\alpha$  denotes the degrees of freedom of the Student's  $t$ -distribution.

### Restricted Boltzmann Machine

In this section, a short introduction to RBM [22,23] is given.

An RBM is a two-layer stochastic NN. This network consists of a visible, or input, layer (visible nodes  $\mathbf{v}$ ) and a hidden layer (hidden nodes  $\mathbf{h}$ ). Values of the nodes are normally Bernoulli-distributed. Each visible node is connected to all hidden nodes using weighted connections, but there are no intra-layer connections. The structure of the RBM is illustrated in Figure 1.

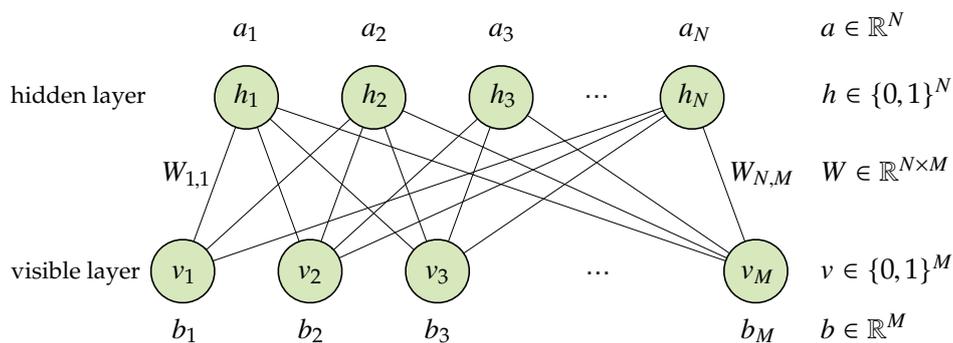


Figure 1. Structure of the RBM.

Boltzmann distribution is specified by the energy function  $E(\mathbf{v}, \mathbf{h})$ , and this distribution gives the joint distribution over all nodes,  $P(\mathbf{v}, \mathbf{h})$ :

$$E(\mathbf{v}, \mathbf{h}) = - \sum_{i,j} W_{ij} v_i h_j - \sum_i b_i v_i - \sum_j a_j h_j,$$

$$P(\mathbf{v}, \mathbf{h}) = \frac{\exp(-E(\mathbf{v}, \mathbf{h}))}{\sum_{\mathbf{v}, \mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}))},$$

where  $W_{ij}$  is the weight of the connection between a visible node  $v_i$  and a hidden node  $h_j$ ; and  $b_i$  and  $a_j$  are the biases of visible and hidden nodes, respectively. Moreover, conditional probabilities  $P(v_i = 1 | \mathbf{h})$  and  $P(h_j = 1 | \mathbf{v})$  are given by the sigmoid function:

$$P(v_i = 1 | \mathbf{h}) = \frac{1}{1 + \exp(-\sum_j W_{ij} h_j - b_i)}, \quad (1)$$

$$P(h_j = 1 | \mathbf{v}) = \frac{1}{1 + \exp(-\sum_i W_{ij} v_i - a_j)}. \quad (2)$$

The RBM can calculate the values of visible nodes from the values of hidden nodes by Equation (1); similarly, the RBM can calculate the values of hidden nodes from the visible nodes by Equation (2).

The model parameters  $W$ ,  $b$ , and  $a$  are learned so that the marginal distribution over the visible nodes under model  $P_{\text{model}}(\mathbf{v})$  is close to the true distribution of data,  $P_{\text{data}}(\mathbf{v})$ . In particular, the RBM uses the Kullback–Leibler (KL) divergence to measure the distance between the true distribution  $P_{\text{data}}(\mathbf{v})$  and the distribution based on the model  $P_{\text{model}}(\mathbf{v})$ . The gradient of the KL divergence with respect to  $W_{ij}$  is given by

$$\frac{\partial \text{KL}(P_{\text{data}} || P_{\text{model}})}{\partial W_{ij}} = \mathbb{E}[v_i h_j]_{P_{\text{data}}} - \mathbb{E}[v_i h_j]_{P_{\text{model}}},$$

where  $\mathbb{E}[v_i h_j]_{P_{\text{data}}}$  is the expected value under the true distribution, and  $\mathbb{E}[v_i h_j]_{P_{\text{model}}}$  is the expected value under the model distribution.

However, the model expectation,  $\mathbb{E}[v_i h_j]_{P_{\text{model}}}$ , cannot be computed analytically. To avoid computing the model, we follow an approximation: the gradient of a slightly different objective function that is called contrastive divergence (CD) [24]. The CD measures how the model distribution gets away from the true distribution of data through  $\text{KL}(P_{\text{data}} || P_{\text{model}}) - \text{KL}(P_1 || P_{\text{model}})$ , where  $P_1(\mathbf{v})$  is the distribution over the visible nodes because the RBM can run for one iteration (that is, one Gibbs sweep) when initialized according to the true distribution. Using standard gradient descent techniques, the CD can be minimized efficiently:

$$\mathbb{E}[v_i h_j]_{P_{\text{data}}} - \mathbb{E}[v_i h_j]_{P_1}.$$

The second term,  $\mathbb{E}[v_i h_j]_{P_1}$ , is estimated from the samples obtained using Gibbs sampling.

### 3. Baseline Data: Preprocessing and Clustering

In this section, data preprocessing is presented briefly, because a more detailed description can be found in [19]. The preprocessing has three stages: data integration (Section 3.1), data transformation (Section 3.2), and data reduction (Section 3.3). Subsequently, data are organized in clusters in Section 3.4.

### 3.1. Data Integration: Unfolding and Scaling

The collected data contain different response signals measured by sensors on a vibrating structure in the time domain. Under different structural states, multiple observations of these responses are measured. Then, using the fast Fourier transform (FFT) algorithm, these response signals are transformed into the frequency domain. All these observations in the frequency domain are collected in a matrix that is defined as follows:

$$\mathbf{X} = \left( x_{i,j}^{k,l} \right) = \begin{bmatrix} x_{1,1}^{1,1} & \cdots & x_{1,1}^{1,L} & x_{1,1}^{2,1} & \cdots & x_{1,1}^{2,L} & \cdots & x_{1,1}^{K,1} & \cdots & x_{1,1}^{K,L} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{n_1,1}^{1,1} & \cdots & x_{n_1,1}^{1,L} & x_{n_1,1}^{2,1} & \cdots & x_{n_1,1}^{2,L} & \cdots & x_{n_1,1}^{K,1} & \cdots & x_{n_1,1}^{K,L} \\ x_{1,2}^{1,1} & \cdots & x_{1,2}^{1,L} & x_{1,2}^{2,1} & \cdots & x_{1,2}^{2,L} & \cdots & x_{1,2}^{K,1} & \cdots & x_{1,2}^{K,L} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{n_2,2}^{1,1} & \cdots & x_{n_2,2}^{1,L} & x_{n_2,2}^{2,1} & \cdots & x_{n_2,2}^{2,L} & \cdots & x_{n_2,2}^{K,1} & \cdots & x_{n_2,2}^{K,L} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{1,j}^{1,1} & \cdots & x_{1,j}^{1,L} & x_{1,j}^{2,1} & \cdots & x_{1,j}^{2,L} & \cdots & x_{1,j}^{K,1} & \cdots & x_{1,j}^{K,L} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{n_j,j}^{1,1} & \cdots & x_{n_j,j}^{1,L} & x_{n_j,j}^{2,1} & \cdots & x_{n_j,j}^{2,L} & \cdots & x_{n_j,j}^{K,1} & \cdots & x_{n_j,j}^{K,L} \end{bmatrix} \in \mathcal{M}_{(n_1+\dots+n_j) \times (K \cdot L)}(\mathbb{R}), \quad (3)$$

where  $K \in \mathbb{N}$  is the number of sensors and  $k = 1, \dots, K$  identifies the sensor that is measuring;  $L \in \mathbb{N}$  is the number of components in each signal and  $l = 1, \dots, L$  indicates the  $l$ -th measurement in the frequency domain;  $J \in \mathbb{N}$  is the number of different structural states that are considered and  $j = 1, \dots, J$  represents the structural state that is measured; finally,  $n_j, j = 1, \dots, J$ , is the number of observations per structural state and  $i = 1, \dots, n_j$  is the  $i$ -th observation related to the  $j$ -th structural state.

Matrix  $\mathbf{X}$  in Equation (3) is a particular unfolded version of a 3-dimensional  $(n_1 + \dots + n_j) \times K \times L$  data matrix, where the first dimension is observation, the second dimension is sensor, and the third dimension is time. Numerous approaches have been proposed to handle 3-dimensional matrices. The most widely adopted approaches are based on the unfolding of these matrices. There are six alternative ways of arranging a 3-dimensional data matrix [20] that affect the performance of the overall strategy, and we have considered type  $E$  in this work, because type- $E$  unfolding simplifies the study of variability among samples.

Matrix  $\mathbf{X}$  in Equation (3) is rescaled through MCGS [21] because of the different magnitudes and scales in the measurements.

### 3.2. Data Transformation

Data transformation means applying a particular mathematical function. The transformation that we apply in this study is PCA, because the final aim is dimensionality reduction. We build the PCA model,  $\mathbf{P}$ , so that the normalized data  $\mathbf{X}$  using MCGS,  $\check{\mathbf{X}}$ , are transformed to the projected data  $\mathbf{T} = \check{\mathbf{X}}\mathbf{P}$ . Notably, matrices  $\check{\mathbf{X}}$  and  $\mathbf{T}$  have equal dimensions; hence, no reduction is performed at this stage.

### 3.3. Data Reduction: PCA and P-t-SNE

In this work, we use two methods of data reduction. On the one hand, we apply PCA to represent the normalized matrix  $\check{\mathbf{X}}$  in a new space with reduced dimensions and without a significant loss of information. On the other hand, we apply P-t-SNE as a 2-dimensional representation technique. These two approaches are combined to reduce the data complexity, computational effort, and time.

### 3.4. Clustering Effect

The first dimensionality reduction is performed using PCA. Specifically, for  $n = n_1 + \dots + n_J$  observations, the rows of matrix  $\mathbf{X}$  in Equation (3) under  $J$  different structural states are projected and transformed into a lower-dimensional space.

Next, the second dimensionality reduction is applied to the projected and transformed data using  $P$ - $t$ -SNE. The aim is to find a collection of 2-dimensional points that represent the projected and transformed data by PCA with no explicit loss of information and preserve the local structure of this dataset. After the application of  $P$ - $t$ -SNE, we expect to observe  $J$  clusters related to  $J$  different structural states.

As mentioned at the beginning of Section 3, see [19] for more details on these stages.

## 4. Structure to Diagnose: Damage Detection and Classification Procedure

In this section, we describe the vibration-based damage detection and classification procedure to diagnose a new structure.

For damage detection and classification, a single observation of the current structure is required to diagnose it. The collected data are composed of different response signals measured by  $K$  sensors and  $L$  components in each signal, as in Equation (3). When these measures are obtained in the frequency domain, we build a new data vector  $\mathbf{z}$ :

$$\mathbf{z}^\top = \left[ z^{1,1} \quad \dots \quad z^{1,L} \mid z^{2,1} \quad \dots \quad z^{2,L} \mid \dots \mid z^{K,1} \quad \dots \quad z^{K,L} \right] \in \mathbb{R}^{K \cdot L}.$$

### 4.1. Scaling (MCGS)

First, we have to scale the row vector  $\mathbf{z}^\top$  to define a scaled row vector  $\check{\mathbf{z}}^\top$ :

$$\check{z}^{k,l} = \frac{z^{k,l} - \mu^{k,l}}{\sigma^k}, \quad k = 1, \dots, K, \quad l = 1, \dots, L, \quad (4)$$

where  $\mu^{k,l}$  is the arithmetic mean of all the elements in the  $[(k-1)L + l]$ -th column of matrix  $\mathbf{X}$  in Equation (3) (that is, the  $l$ -th column of the  $k$ -th sensor) and  $\sigma^k$  is the standard deviation of all measurements of the  $k$ -th sensor relative to the mean value  $\mu^k$  (the arithmetic mean of all the measurements of the  $k$ -th sensor).

### 4.2. Projection (PCA)

The scaled row vector  $\check{\mathbf{z}}^\top \in \mathbb{R}^{K \cdot L}$  is projected into the space spanned by the first  $\ell$  principal components in  $\mathbf{P}_\ell$  through the vector-to-matrix multiplication:

$$\mathbf{x}^{n+1} = \check{\mathbf{z}}^\top \cdot \mathbf{P}_\ell \in \mathbb{R}^\ell.$$

Notably, the vector containing the data of the structure to be diagnosed initially has dimension  $K \cdot L$  but later dimension  $\ell$ . We add this new point to the projected and transformed data by PCA ( $\mathcal{X} = \{\mathbf{x}^1, \dots, \mathbf{x}^n\} \subset \mathbb{R}^\ell$ ) to define a new set:

$$\mathcal{X}' = \{\mathbf{x}^1, \dots, \mathbf{x}^n\} \cup \{\mathbf{x}^{n+1}\} = \{\mathbf{x}^1, \dots, \mathbf{x}^n, \mathbf{x}^{n+1}\} \subset \mathbb{R}^\ell, \quad (5)$$

where

$$\mathbf{x}^i = \mathbf{e}_i^\top \check{\mathbf{X}} \mathbf{P}_\ell, \quad i = 1, \dots, n$$

and  $\mathbf{e}_i \in \mathbb{R}^n$  is the  $i$ -th element of the canonical basis. The network is trained with  $\{\mathbf{x}^1, \dots, \mathbf{x}^n\}$ . Then,  $\{\mathbf{x}^{n+1}\}$  will be passed through the trained network.

### 4.3. P-t-SNE and Final Classification

Finally, we apply P-t-SNE to the  $\ell$ -dimensional set  $\mathcal{X}'$  in Equation (5) to find a collection of 2-dimensional map points  $\mathcal{Y}' = \{\mathbf{y}^1, \dots, \mathbf{y}^n, \mathbf{y}^{n+1}\} \subset \mathbb{R}^2$  that represent the original set  $\mathcal{X} = \{\mathbf{x}^1, \dots, \mathbf{x}^n\} \subset \mathbb{R}^\ell$  (the data projected and transformed by PCA) with no explicit loss of information and retaining the local structure of this set. Furthermore, the map point  $\mathbf{y}^{n+1}$ , associated with the data point  $\mathbf{x}^{n+1}$ , is included. That is, the embedded data are constructed applying the trained network: input  $\mathcal{X}'$  and output  $\mathcal{Y}'$ . We expect to observe the same  $J$  clusters related to the  $J$  different structural states.

For each cluster, we calculate its centroid: the mean of the values of the data points in the cluster. For instance, the centroid associated with the first structural state is

$$\mathcal{Y}'_1 := \frac{1}{n_1} \sum_{i=1}^{n_1} \mathbf{y}^i = \frac{\mathbf{y}^1 + \dots + \mathbf{y}^{n_1}}{n_1} \in \mathbb{R}^2.$$

In general, the centroid associated with the  $j$ -th structural state,  $j = 1, \dots, J$ , is the 2-dimensional point defined as

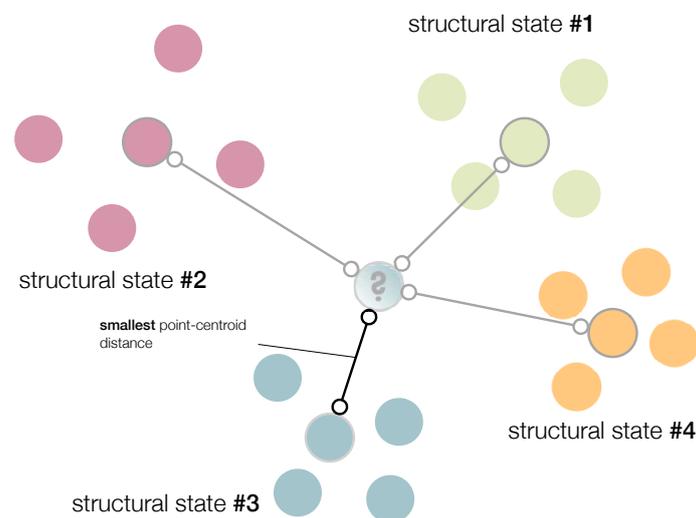
$$\mathcal{Y}'_j := \frac{1}{n_j} \sum_{i=1}^{n_j} \mathbf{y}^{\left(\sum_{m=0}^{j-1} n_m\right)+i} \in \mathbb{R}^2, \quad j = 1, \dots, J, \quad (6)$$

where  $n_0 = 0$ . As a result, the current structure to diagnose is associated with the  $j$ -th structural state if

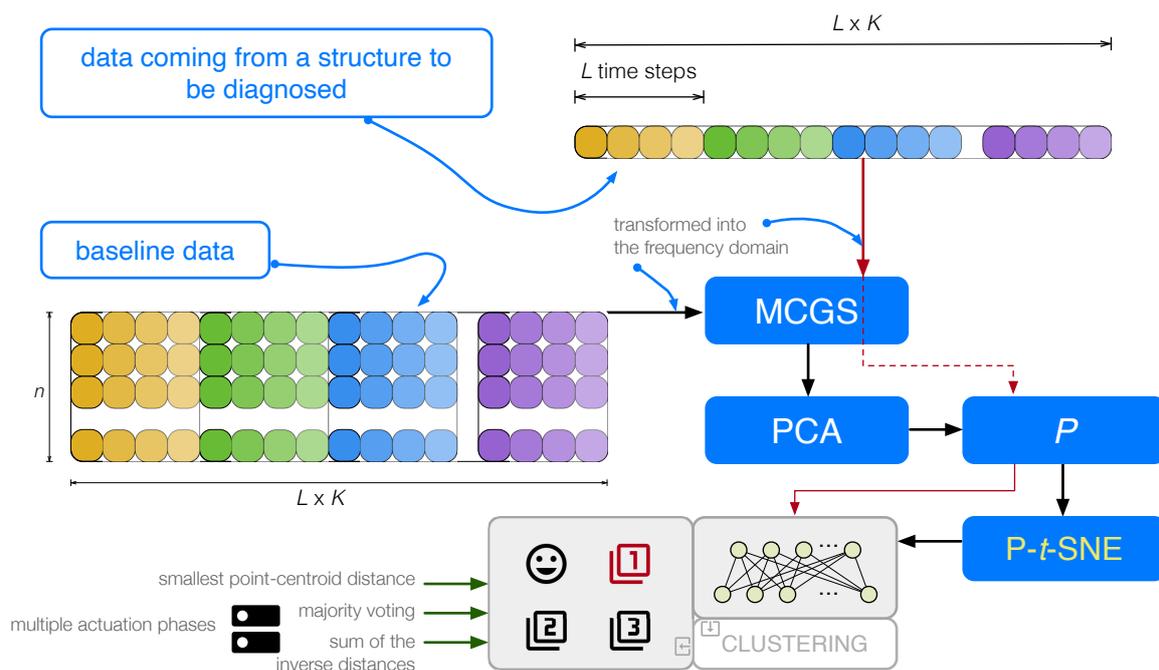
$$j = \arg \min_{j=1, \dots, J} \|\mathcal{Y}'_j - \mathbf{y}^{n+1}\|_2,$$

that is, if the minimum distance between  $\mathbf{y}^{n+1}$  and each centroid corresponds to the Euclidean distance between  $\mathcal{Y}'_j$  and  $\mathbf{y}^{n+1}$ . We call this approach the smallest point-centroid distance (see Figure 2).

The proposed procedure is shown in Figure 3.



**Figure 2.** Current structure to diagnose is associated with the structural state with the smallest point-centroid distance.



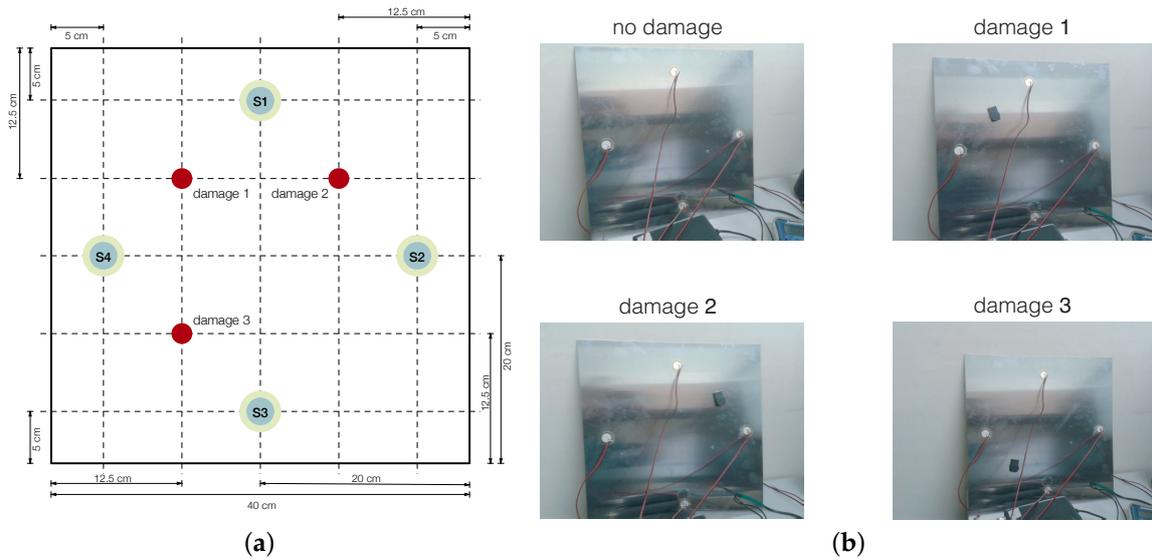
**Figure 3.** Flowchart of the proposed procedure. Data coming from a structure are first scaled by MCGS and then projected into the PCA model. Finally, P-t-SNE is applied to generate the clusters that will be used in the vibration-based detection and classification of structural changes.

## 5. Application of the SHM System on an Aluminum Plate with Four PZTs

In this study, we reuse the structure and experiment from [19]. Therefore, we can use this structure as a benchmark to compare our results.

### 5.1. Structure

A square aluminum plate was manufactured to demonstrate the accuracy of the vibration-based method of damage detection and classification presented in Sections 3 and 4. The dimension of the plate is  $40 \times 40 \times 0.2$  cm. The plate is instrumented with four PZTs and a mass of 17.2916 g is introduced to simulate the damage in a non-destructive way, producing changes in the propagated wave (see Figure 4a). Each PZT can work in two modes: excite the aluminum plate (actuator mode) with a burst signal or detect a time-varying mechanical response (sensor mode). The location of the mass defines each damage, and  $J = 4$  structural states are considered: healthy state and three different types of damage (Figure 4b). The plate is isolated from the vibration and noise in the laboratory (Figure 4b).



**Figure 4.** (a) Aluminum plate with four piezoelectric sensors (S1, S2, S3, and S4); (b) the four structural states considered.

### 5.2. Scenarios and Actuation Phases

Unlike [19], in this study, the experimental setup includes only two scenarios (the two extreme cases) to determine the performance of the proposed method:

- **Scenario 1.** The signals are obtained using a short wire (0.5 m) from the digitizer to the piezoelectric sensors. Next, the Savitzky–Golay [25] algorithm is used to filter these signals after adding white Gaussian noise. This filter is used to smoothen the data.
- **Scenario 2.** The signals are obtained using a long wire (2.5 m) from the digitizers to the piezoelectric sensors. Signals are also filtered with the Savitzky–Golay algorithm.

As discussed in Section 5.1, there are four PZTs (S1, S2, S3, and S4) that excite the plate and collect the measured signal. This sensor network works in what we call actuation phases. In each actuation phase, a PZT is used as an actuator (which excites the plate with the burst signal), and the rest of the PZTs are used as sensors (which measure signals). Therefore, we have as many actuation phases as sensors: in actuation phase 1, S1 is used as the actuator and the rest of PZTs are used as sensors; in actuation phase 2, S2 is used as the actuator and the rest of PZTs are used as sensors; and so on.

### 5.3. Data Collection

Given a certain scenario, as the two defined in Section 5.2, four matrices  $\mathbf{X}[\varphi]$ ,  $\varphi = 1, \dots, 4$ , are obtained, one for each actuation phase. Each matrix  $\mathbf{X}[\varphi]$ ,  $\varphi = 1, \dots, 4$ , is constructed as follows:

- $n_1 = n_2 = n_3 = n_4 = 25$  observations are performed for each of the four structural states. Therefore, each matrix  $\mathbf{X}[\varphi]$ ,  $\varphi = 1, \dots, 4$ , contains 100 rows:  $n_1 + n_2 + n_3 + n_4 = 25 \cdot 4$ . Specifically, the first 25 rows represent the healthy state, the next 25 are observations with damage 1, and so on.
- For each actuation phase  $\varphi$ ,  $\varphi = 1, \dots, 4$ ,  $K = 3$  PZTs working as sensors are measured during 60000 time instants. Next, these measurements are transformed into the frequency domain. As a result, the number of columns of matrix  $\mathbf{X}[\varphi]$ ,  $\varphi = 1, \dots, 4$ , is equal to  $K \cdot L = 3 \cdot ((60000/2) + 1) = 90003$ .

Therefore, the matrix that collects all observations under the different structural states in the frequency domain is as follows (see Equation (3)); here,  $L = 30001$  and  $J = 4$ :

$$\mathbf{X}[\varphi] = \left( x[\varphi]_{i,j}^{k,l} \right) \in \mathcal{M}_{100 \times 90003}(\mathbb{R}). \quad (7)$$

The damage detection and classification method described in Sections 3 and 4 can be applied to each matrix  $\mathbf{X}[\varphi]$ ,  $\varphi = 1, \dots, 4$ , in Equation (7), leading to one classification per actuation phase. Then, we will use these four classifications to obtain a final decision based on the four actuation phases. This strategy will be detailed in Section 5.5.

#### 5.4. $\kappa$ -Fold Nonexhaustive Leave-p-Out Cross-Validation

The proposed approach is evaluated by comparing test data (the new observations in unknown state under the same conditions) with baseline data (data from the structure under the four different structural states). For that purpose, the  $\kappa$ -fold nonexhaustive leave- $p$ -out cross-validation [19] is used. Hence, the sum of all elements in the confusion matrices that are presented in Section 6 is equal to 400. We use the notation  $\mathfrak{X}$  to represent the matrix that is used as the baseline to build the model. Matrix  $\mathfrak{X}$  has 20 rows: five for each structural state.

#### 5.5. Damage Detection and Classification

The strategy for damage detection and classification is as follows: the classification will be based on the four matrices  $\mathbf{X}[1], \mathbf{X}[2], \mathbf{X}[3]$ , and  $\mathbf{X}[4]$ , defined in Equation (7), with  $\kappa$ -fold nonexhaustive leave- $p$ -out cross-validation. Each actuation phase will cast a vote that will determine the final classification.

In this strategy, the next steps are followed:

- **Step 1.** The data matrix  $\mathfrak{X}$  is scaled by MCGS, obtaining matrix  $\check{\mathfrak{X}}$ .
- **Step 2.** PCA is applied to matrix  $\check{\mathfrak{X}}$ , obtaining the PCA model  $\mathbf{P}$ .
- **Step 3.** The reduced PCA model  $\mathbf{P}_\ell$  is chosen such that the proportion of variance explained is at least 95%.
- **Step 4.** An observation  $\mathbf{z}^\top \in \mathbb{R}^{3 \cdot 30001} = \mathbb{R}^{90003}$  of the current structure-to-diagnose is needed. Then,  $\mathbf{z}^\top$  is scaled by Equation (4) to obtain  $\check{\mathbf{z}}^\top$ , which is projected into the space spanned by the first  $\ell$  principal components in  $\mathbf{P}_\ell$ .
- **Step 5.** Dataset  $\mathcal{X}' = \{\mathbf{x}^1, \dots, \mathbf{x}^{20}\} \cup \{\mathbf{x}^{21}\} = \{\mathbf{x}^1, \dots, \mathbf{x}^{20}, \mathbf{x}^{21}\} \subset \mathbb{R}^\ell$  is defined by Equation (5). The network is trained with  $\{\mathbf{x}^1, \dots, \mathbf{x}^{20}\}$ . Then,  $\{\mathbf{x}^{21}\}$  will be passed through the trained network.
- **Step 6.**  $P$ - $t$ -SNE is applied to  $\mathcal{X}'$  to find a set of 2-dimensional points:  $\mathcal{Y}' = \{\mathbf{y}^1, \dots, \mathbf{y}^{20}, \mathbf{y}^{21}\} \subset \mathbb{R}^2$ . Thus, the embedded data are constructed using the trained network: input  $\mathcal{X}'$  and output  $\mathcal{Y}'$ .
- **Step 7.**  $J = 4$  clusters are obtained, one per structural state. These clusters are formed by the 2-dimensional points:  $\{\mathbf{y}^1, \dots, \mathbf{y}^5\} \subset \mathcal{Y}'$ , related to the healthy state;  $\{\mathbf{y}^6, \dots, \mathbf{y}^{10}\} \subset \mathcal{Y}'$ , related to damage 1;  $\{\mathbf{y}^{11}, \dots, \mathbf{y}^{15}\} \subset \mathcal{Y}'$ , related to damage 2; and  $\{\mathbf{y}^{16}, \dots, \mathbf{y}^{20}\} \subset \mathcal{Y}'$ , related to damage 3. Centroid  $\mathcal{Y}'_j$ ,  $j = 1, \dots, J$ , associated with the  $j$ -th structural state is calculated by Equation (6).
- **Step 8.** With the information given by the four actuation phases, the structure that must be diagnosed is finally classified considering two approaches: majority voting and sum of the inverse distances. For details of both approaches, see [19].

## 6. Results

In this section, confusion matrices summarize the results of the damage detection and classification method presented in Sections 3 and 4 and detailed in Sections 5.3–5.5. The results for each scenario are shown in different subsections. Four different structural states are considered in both scenarios:

- The healthy state ( $D0$ )—the aluminum plate with no damage;
- Three states with damage ( $D1$ ,  $D2$ , and  $D3$ )—the aluminum plate with an added mass at the positions indicated in Figure 4.

Five iterations ( $\kappa = 5$ ) of a nonexhaustive leave- $p$ -out cross-validation, with  $p = 80$ , are performed to validate the damage detection and classification method. At each iteration, 80 observations are considered: 20 observations per structural state ( $D0, D1, D2$ , and  $D3$ ). Therefore, the sum of all elements in the confusion matrices is equal to  $5 \cdot 80 = 400$ .

Two different confusion matrices are shown for each of the two scenarios:

- **Majority voting.** The damage detection and classification method is applied to the four matrices  $\mathbf{X}[\varphi]$ ,  $\varphi = 1, \dots, 4$ . Each actuation phase emits a vote, and the final classification is made by the majority voting [19].
- **Sum of the inverse distances.** The damage detection and classification method is applied to the four matrices  $\mathbf{X}[\varphi]$ ,  $\varphi = 1, \dots, 4$ . Each actuation phase emits a vote, and the final classification is made using the maximum sum of the inverse distances [19].

Finally, we have included the confusion matrices for  $t$ -SNE, to compare its performance with P- $t$ -SNE.

### 6.1. Scenario 1

In this section, we describe the results for **Scenario 1** from Section 5.2. Tables 1 and 2 show the four confusion matrices. The green background cells correspond to observations that are correctly classified, while the red background cells represent the misclassifications. The color intensity (green or red) is related to the proportion of correct or wrong decisions.

With P- $t$ -SNE and the majority voting (Table 1), the overall accuracy is very good. Specifically, 398 out of 400 observations have been correctly classified, which corresponds to the overall accuracy of 99.5%. With  $t$ -SNE, the overall accuracy is 100% (Table 1).

Using the sum of the inverse distances (Table 2) for the P- $t$ -SNE-based damage detection and classification, 399 out of 400 observations have been correctly classified; hence, the overall accuracy is 99.75%. For  $t$ -SNE, 400 out of 400 observations have been correctly classified (100% accuracy).

Furthermore, other metrics are calculated. The most common metrics for choosing the best solution in a binary classification problem are as follows:

- *Accuracy*, defined as the proportion of true results among the total number of cases examined.
- *Precision or positive predictive value (PPV)* that attempts to answer the question “what proportion of positive identifications is actually correct?”.
- *Sensitivity or true positive rate (TPR)* that measures the proportion of actual positives that are correctly identified as such.
- *F<sub>1</sub> score*, defined as the harmonic mean of PPV and TPR.
- *Specificity or true negative rate (TNR)* that measures the proportion of actual negatives that are correctly identified as such.

These metrics are easy to calculate for binary and multiclass classification problems [26]. When the classification problem is multiclass, as in the current work, according to [27,28], the result is the average obtained by adding the result of each class and dividing over the total number of classes.

In all cases (P- $t$ -SNE,  $t$ -SNE, the majority voting, and the sum of the inverse distances), these five metrics vary between 99.5% and 100% (Tables 3 and 4).

As can be seen, both parametric and non-parametric approaches obtain practically the same results. However, P- $t$ -SNE dramatically reduces the processing time: it decreases from 40 min and 15 s ( $t$ -SNE) to 2 min and 34 s (P- $t$ -SNE) on an Intel Core i7 4.20 GHz computer with 32 GB RAM. Using P- $t$ -SNE, a decision is made in just a few milliseconds. The reduced time (2 min and 34 s) includes both the pretraining and the fine-tuning of the NN, as well as the classification of the current state of the structure. The total computational cost of P- $t$ -SNE is reduced by approximately 94% compared with  $t$ -SNE.

**Table 1.** Confusion matrix of the application of P-*t*-SNE and *t*-SNE damage detection and classification method presented in Sections 3 and 4: scenario 1 and the majority voting approach.

True \ Predicted	P- <i>t</i> -SNE				<i>t</i> -SNE			
	D0	D1	D2	D3	D0	D1	D2	D3
D0	100	0	0	0	100	0	0	0
D1	0	100	0	0	0	100	0	0
D2	0	0	100	0	0	0	100	0
D3	0	0	2	98	0	0	0	100

<sup>1</sup> D0: healthy state of the structure; D1, D2, and D3: added masses at the positions indicated in Figure 4.

**Table 2.** Confusion matrix of the application of P-*t*-SNE and *t*-SNE damage detection and classification method presented in Sections 3 and 4: scenario 1 and the sum of the inverse distances approach.

True \ Predicted	P- <i>t</i> -SNE				<i>t</i> -SNE			
	D0	D1	D2	D3	D0	D1	D2	D3
D0	100	0	0	0	100	0	0	0
D1	0	100	0	0	0	100	0	0
D2	0	0	99	1	0	0	100	0
D3	0	0	0	100	0	0	0	100

<sup>1</sup> D0: healthy state of the structure; D1, D2, and D3: added masses at the positions indicated in Figure 4.

**Table 3.** Accuracy, PPV, TPR,  $F_1$  score, and TNR of the application of P-*t*-SNE and *t*-SNE damage detection and classification method presented in Sections 3 and the 4: scenario 1 and the majority voting approach.

	P- <i>t</i> -SNE	<i>t</i> -SNE
Accuracy	99.8%	100.0%
PPV	99.5%	100.0%
TPR	99.5%	100.0%
$F_1$ score	99.5%	100.0%
TNR	99.8%	100.0%

**Table 4.** Accuracy, PPV, TPR,  $F_1$  score, and TNR of the application of P-*t*-SNE and *t*-SNE damage detection and classification method presented in Sections 3 and 4: scenario 1 and the sum of the inverse distances approach.

	P- <i>t</i> -SNE	<i>t</i> -SNE
Accuracy	99.9%	100.0%
PPV	99.8%	100.0%
TPR	99.8%	100.0%
$F_1$ score	99.7%	100.0%
TNR	99.9%	100.0%

## 6.2. Scenario 2

In this section, we describe the results for Scenario 2 from Section 5.2. Tables 5 and 6 show the four confusion matrices.

With P-*t*-SNE and the majority voting (Table 5), the overall accuracy is also very good. Specifically, 398 out of 400 observations have been correctly classified; this corresponds to the overall accuracy of 99.5%. With *t*-SNE and the majority voting, the overall accuracy is 100% (Table 5).

Using the maximum sum of the inverse distances to take a final decision (Table 6) with P-*t*-SNE, 399 out of 400 observations have been correctly classified (the overall accuracy of 99.75%). With *t*-SNE, 400 out of 400 observations have been correctly classified (the overall accuracy of 100%).

In the non-parametric approach, using the majority voting and the sum of the inverse distances, the five metrics presented in Section 6.1 achieve 100%. In contrast, in the parametric approach, these metrics slightly decrease (between 0.1% and 0.5%, see Tables 7 and 8).

Again, as in scenario 1, parametric and non-parametric methods obtain similar results. However, as before, the P-*t*-SNE approach dramatically reduces the processing time: from 42 min and 1 s (*t*-SNE) to 2 min and 32 s (P-*t*-SNE). As in the previous scenario, the total computational cost of P-*t*-SNE is reduced by approximately 94% compared with *t*-SNE.

**Table 5.** Confusion matrix of the application of P-*t*-SNE and *t*-SNE damage detection and classification method presented in Sections 3 and 4: scenario 2 and the majority voting approach.

True \ Predicted	P- <i>t</i> -SNE				<i>t</i> -SNE			
	D0	D1	D2	D3	D0	D1	D2	D3
D0	100	0	0	0	100	0	0	0
D1	1	99	0	0	0	100	0	0
D2	1	0	99	0	0	0	100	0
D3	0	0	0	100	0	0	0	100

<sup>1</sup> D0: healthy state of the structure; D1, D2, and D3: added masses at the positions indicated in Figure 4.

**Table 6.** Confusion matrix of the application of P-*t*-SNE and *t*-SNE damage detection and classification method presented in Sections 3 and 4: scenario 2 and the sum of the inverse distances approach.

True \ Predicted	P- <i>t</i> -SNE				<i>t</i> -SNE			
	D0	D1	D2	D3	D0	D1	D2	D3
D0	100	0	0	0	100	0	0	0
D1	1	99	0	0	0	100	0	0
D2	0	0	100	0	0	0	100	0
D3	0	0	0	100	0	0	0	100

<sup>1</sup> D0: healthy state of the structure; D1, D2, and D3: added masses at the positions indicated in Figure 4.

**Table 7.** Accuracy, PPV, TPR,  $F_1$  score, and TNR of the application of P-*t*-SNE and *t*-SNE damage detection and classification method presented in Sections 3 and 4: scenario 2 and the majority voting approach.

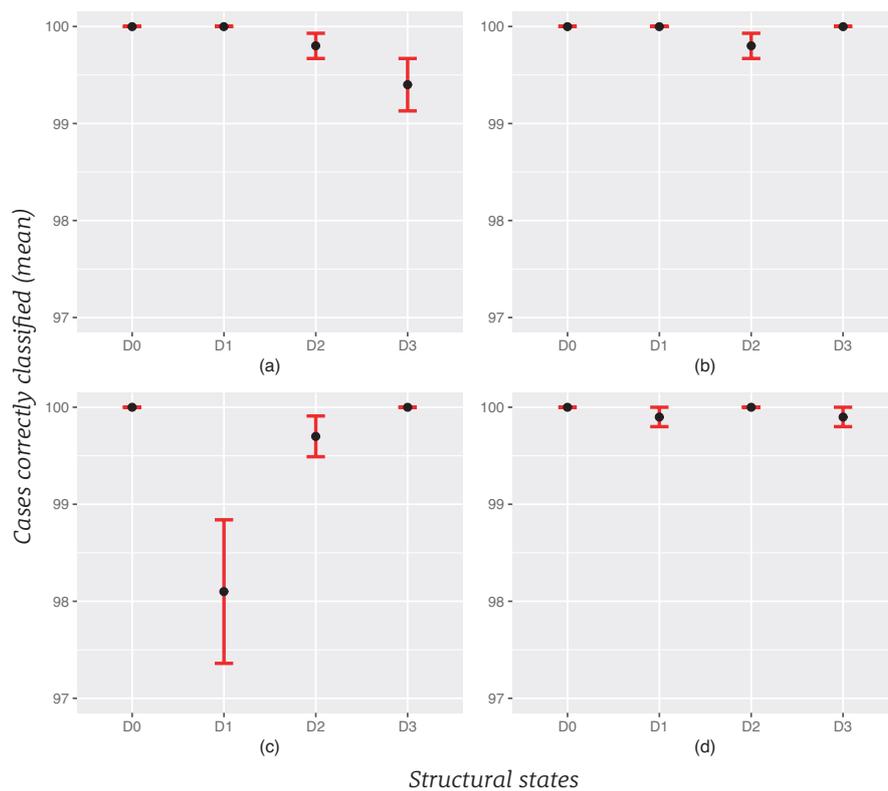
	P- <i>t</i> -SNE	<i>t</i> -SNE
Accuracy	99.8%	100.0%
PPV	99.5%	100.0%
TPR	99.5%	100.0%
$F_1$ score	99.5%	100.0%
TNR	99.8%	100.0%

**Table 8.** Accuracy, PPV, TPR,  $F_1$  score, and TNR of the application of P-*t*-SNE and *t*-SNE damage detection and classification method presented in Sections 3 and 4: scenario 2 and the sum of the inverse distances approach.

	P- <i>t</i> -SNE	<i>t</i> -SNE
Accuracy	99.9%	100.0%
PPV	99.8%	100.0%
TPR	99.8%	100.0%
$F_1$ score	99.7%	100.0%
TNR	99.9%	100.0%

### 6.3. Repeatability

We use error bars to give the reader a general idea of the uncertainty in the results. Figure 5 shows the mean of each structural state with error bars representing the standard error. As it can be seen both in Figure 5 and in Tables 9–12, the standard error is very small, when we repeat the procedure 10 times.



**Figure 5.** Repeatability of the SHM strategy (10 times), graphs with error bars: (a) scenario 1 and majority voting approach; (b) scenario 1 and sum of the inverse distances approach; (c) scenario 2 and majority voting approach; (d) scenario 2 and sum of the inverse distances approach. *D0*: healthy state of the structure; *D1*, *D2*, and *D3*: added masses at the positions indicated in Figure 4.

**Table 9.** Repeatability of the SHM strategy (10 times): scenario 1 and majority voting approach. *D0*: healthy state of the structure; *D1*, *D2*, and *D3*: added masses at the positions indicated in Figure 4.

	<i>D0</i>	<i>D1</i>	<i>D2</i>	<i>D3</i>
Mean	100.00	100.00	99.80	99.40
Standard deviation	0.00	0.00	0.42	0.84
Standard error	0.00	0.00	0.13	0.27

**Table 10.** Repeatability of the SHM strategy (10 times): scenario 1 and sum of the inverse distances approach. *D0*: healthy state of the structure; *D1*, *D2*, and *D3*: added masses at the positions indicated in Figure 4.

	<i>D0</i>	<i>D1</i>	<i>D2</i>	<i>D3</i>
Mean	100.00	100.00	99.80	100.00
Standard deviation	0.00	0.00	0.42	0.00
Standard error	0.00	0.00	0.13	0.00

**Table 11.** Repeatability of the SHM strategy (10 times): scenario 2 and majority voting approach. *D0*: healthy state of the structure; *D1*, *D2*, and *D3*: added masses at the positions indicated in Figure 4.

	<i>D0</i>	<i>D1</i>	<i>D2</i>	<i>D3</i>
Mean	100.00	98.10	99.70	100.00
Standard deviation	0.00	2.33	0.67	0.00
Standard error	0.00	0.74	0.21	0.00

**Table 12.** Repeatability of the SHM strategy (10 times): scenario 2 and sum of the inverse distances approach. *D0*: healthy state of the structure; *D1*, *D2*, and *D3*: added masses at the positions indicated in Figure 4.

	<i>D0</i>	<i>D1</i>	<i>D2</i>	<i>D3</i>
Mean	100.00	99.90	100.00	99.90
Standard deviation	0.00	0.32	0.00	0.32
Standard error	0.00	0.10	0.00	0.10

## 7. Conclusions

In this paper, we proposed an SHM strategy for the detection and classification of structural changes combining PCA and P-*t*-SNE. We evaluated the proposed method with experimental data. The obtained results show that its performance is very good, given its high classification accuracy. In addition, we have compared the parametric version of *t*-SNE with the non-parametric version.

According to the results shown in Sections 6.1 and 6.2, the performance is very satisfactory and similar in both approaches: P-*t*-SNE and *t*-SNE. However, in terms of processing time, it is better to make a decision considering the P-*t*-SNE-based damage detection and classification rather than working with the *t*-SNE-based method: although the non-parametric approach slightly outperforms the parametric approach, the parametric approach can reduce the total computational cost by approximately 94%. Hence, P-*t*-SNE can classify a current structure in just a few milliseconds. This is the first indication that P-*t*-SNE is better compared with the *t*-SNE. Other advantages of using P-*t*-SNE are as follows:

1. P-*t*-SNE can handle large-scale datasets, while *t*-SNE can only handle datasets with a size not greater than the order of thousands.
2. The *t*-SNE method requires an extremely large computational cost for the optimization: to map a new data sample, the optimization has to run for the whole dataset again. However, P-*t*-SNE can learn from the training data and be applied when a new observation arises; hence, it can work with real-time observations. Therefore, the parametric approach can make inferences about new samples to be diagnosed without having to recalculate everything; the model predicts on out-of-sample data.

Based on the foregoing, and seeing the strong performance of the P-*t*-SNE-based approach, we conclude that it is better to work with the parametric version of *t*-SNE than with the non-parametric version.

Many possible fields of application exist. For example, in aeronautics, parts of an airplane (wings or fuselage) can be simulated with similar aluminum plates; for wind turbines, this methodology can be applied to detect damage and faults. In general, if a sensor network can be installed in a structure, and various actuation phases can be defined, the proposed approach can be considered.

In our future work, we contemplate to apply the proposed methodology to handle imbalanced data, as well as to determine its effectiveness in different environmental and operational conditions.

**Author Contributions:** D.A. and F.P. developed the idea and designed the exploration framework; D.A. developed the algorithms; F.P. supervised the results of the validation; D.A. and F.P. drafted the manuscript. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research has been partially funded by the Spanish Agencia Estatal de Investigación (AEI)—Ministerio de Economía, Industria y Competitividad (MINECO), by the Fondo Europeo de Desarrollo Regional (FEDER) through the research project DPI2017-82930-C2-1-R, and by the Generalitat de Catalunya through the research project 2017 SGR 388.

**Acknowledgments:** We thank the Universitat Politècnica de Catalunya (UPC) for predoctoral fellowship (to David Agis).

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

## Abbreviations

The following abbreviations are used in this manuscript:

CD	contrastive divergence
FFT	fast Fourier transform
ISOMAP	isometric mapping
KL	Kullback–Leibler
LDA	linear discriminant analysis
MCGS	mean-centered group scaling
NN	neural network
PCA	principal component analysis
PPV	positive predictive value
P- <i>t</i> -SNE	parametric <i>t</i> -distributed stochastic neighbor embedding
PZT	piezoelectric transducers
RBM	restricted Boltzmann machine
SHM	structural health monitoring
TNR	true negative rate
TPR	true positive rate
<i>t</i> -SNE	<i>t</i> -distributed stochastic neighbor embedding

## References

- Worden, K.; Cross, E. On switching response surface models, with applications to the structural health monitoring of bridges. *Mech. Syst. Sig. Process.* **2018**, *98*, 139–156. [\[CrossRef\]](#)
- Liu, Y.; Kim, S.B.; Chattopadhyay, A.; Doyle, D. Application of system-identification techniques to health monitoring of on-orbit satellite boom structures. *J. Spacecraft Rockets* **2011**, *48*, 589–598. [\[CrossRef\]](#)
- Sohn, H.; Farrar, C.R.; Hunter, N.F.; Worden, K. Structural health monitoring using statistical pattern recognition techniques. *J. Dyn. Sys. Meas. Control* **2001**, *123*, 706–711. [\[CrossRef\]](#)
- Gui, G.; Pan, H.; Lin, Z.; Li, Y.; Yuan, Z. Data-driven support vector machine with optimization techniques for structural health monitoring and damage detection. *KSCE J. Civ. Eng.* **2017**, *21*, 523–534. [\[CrossRef\]](#)
- Jimenez, L.O.; Landgrebe, D.A. Supervised classification in high-dimensional space: geometrical, statistical, and asymptotical properties of multivariate data. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **1998**, *28*, 39–54. [\[CrossRef\]](#)
- Tibaduiza, D.; Mujica, L.; Rodellar, J. Damage classification in structural health monitoring using principal component analysis and self-organizing maps. *Struct. Control Health Monit.* **2013**, *20*, 1303–1316. [\[CrossRef\]](#)
- Caggiano, A. Tool wear prediction in Ti-6Al-4V machining through multiple sensor monitoring and PCA features pattern recognition. *Sensors* **2018**, *18*, 823. [\[CrossRef\]](#) [\[PubMed\]](#)
- Sharma, A.; Paliwal, K.K.; Onwubolu, G.C. Class-dependent PCA, MDC and LDA: A combined classifier for pattern classification. *Pattern Recognit.* **2006**, *39*, 1215–1229. [\[CrossRef\]](#)
- Elvira, M.; Iáñez, E.; Quiles, V.; Ortiz, M.; Azorín, J.M. Pseudo-online BMI based on EEG to detect the appearance of sudden obstacles during walking. *Sensors* **2019**, *19*, 5444. [\[CrossRef\]](#) [\[PubMed\]](#)
- Jeong, M.; Choi, J.H.; Koh, B.H. Isomap-based damage classification of cantilevered beam using modal frequency changes. *Struct. Control Health Monit.* **2014**, *21*, 590–602. [\[CrossRef\]](#)
- Ullah, S.; Jeong, M.; Lee, W. Nondestructive inspection of reinforced concrete utility poles with ISOMAP and random forest. *Sensors* **2018**, *18*, 3463. [\[CrossRef\]](#) [\[PubMed\]](#)
- Van Der Maaten, L.; Hinton, G.E. Visualizing data using *t*-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
- Zhang, X.; Gou, L.; Li, Y.; Feng, J.; Jiao, L. Gaussian process latent variable model based on immune clonal selection for SAR target feature extraction and recognition. *J. Infrared Millim. Waves* **2013**, *32*, 231–236. [\[CrossRef\]](#)
- Kebede, T.M.; Djaneye-Boundjou, O.; Narayanan, B.N.; Ralescu, A.; Kapp, D. Classification of malware programs using autoencoders based deep learning architecture and its application to the microsoft malware classification challenge (big 2015) dataset. In Proceedings of the 2017 IEEE National Aerospace and Electronics Conference (NAECON), Dayton, OH, USA, 27–30 June 2017.

15. Li, Y.; Wang, Y.; Zi, Y.; Zhang, M. An enhanced data visualization method for diesel engine malfunction classification using multi-sensor signals. *Sensors* **2015**, *15*, 26675–26693. [[CrossRef](#)]
16. Balamurali, M.; Melkumyan, A. *t*-SNE based visualisation and clustering of geological domain. In Proceedings of the International Conference on Neural Information Processing, Kyoto, Japan, 16–21 October 2016.
17. Peng, Z.; Cao, C.; Liu, Q.; Pan, W. Human walking pattern recognition based on KPCA and SVM with ground reflex pressure signal. *Math. Probl. Eng.* **2013**, *2013*. [[CrossRef](#)]
18. Van Der Maaten, L. Learning a parametric embedding by preserving local structure. In Proceedings of the 12th International Conference on Artificial Intelligence and Statistics, Clearwater, FL, USA, 16–19 April 2009.
19. Agis, D.; Pozo, F. A frequency-based approach for the detection and classification of structural changes using *t*-SNE. *Sensors* **2019**, *19*, 5097. [[CrossRef](#)]
20. Westerhuis, J.A.; Kourti, T.; MacGregor, J.F. Comparing alternative approaches for multivariate statistical analysis of batch process data. *J. Chemom. A J. Chemom. Soc.* **1999**, *13*, 397–413. [[CrossRef](#)]
21. Pozo, F.; Vidal, Y.; Salgado, Ó. Wind turbine condition monitoring strategy through multiway PCA and multivariate inference. *Energies* **2018**, *11*, 749. [[CrossRef](#)]
22. Smolensky, P. Information processing in dynamical systems: foundations of harmony theory. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Foundations*; Rumelhart, D.E., McClelland, J.L., Eds.; MIT Press: Cambridge, MA, USA, 1986; Volume 1, pp. 194–281.
23. Hinton, G.E. A practical guide to training restricted Boltzmann machines. In *Neural Networks: Tricks of the Trade*; Springer: Berlin, Germany, 2012; pp. 599–619.
24. Hinton, G.E. Training products of experts by minimizing contrastive divergence. *Neural Comput.* **2002**, *14*, 1771–1800. [[CrossRef](#)]
25. Orfanidis, S.J. *Introduction to Signal Processing*; Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 1995.
26. Hossin, M.; Sulaiman, M. A review on evaluation metrics for data classification evaluations. *Int. J. Data Min. Knowl. Manage. Process* **2015**, *5*, 1–11.
27. Krüger, F. Activity, Context, and Plan Recognition with Computational Causal Behaviour Models. Ph.D. Thesis, University of Rostock, Mecklenburg, Germany, 26 September 2016.
28. Hameed, N.; Hameed, F.; Shabut, A.; Khan, S.; Cirstea, S.; Hossain, A. An intelligent computer-aided scheme for classifying multiple skin lesions. *Computers* **2019**, *8*, 62. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).