



# Frequency domain CNN and dissipated energy approach for damage detection in building structures

Mario Lopez-Pacheco<sup>1</sup> · Jesús Morales-Valdez<sup>2</sup> · Wen Yu<sup>1</sup>

Published online: 18 April 2020  
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

## Abstract

Recent developments tools and techniques for structural health monitoring allow the design of early warning systems for the damage diagnosis and structural assessment. Most methods to damage detection involve vibration data analysis by using identification systems that generally require a mathematical model and much information about the system, such as parameters and states that are mostly unknown. In this paper, a novel frequency domain convolutional neural network (FDCNN) proposed aims to design an identification system for damage detection based on Bouc–Wen hysteretic model. FDCNN, unlike other works, only requires acceleration measurements for damage diagnosis that are very sensitive to environmental noise. In contrast to neural network (NN) and time domain convolutional neural network, FDCNN reduces the computational time required for the learning stage and adds robustness against noise in data. The FDCNN includes random filters in the frequency domain to avoid measurement noise using a spectral pooling operation, which is useful when the system bandwidth is unknown. Incorrect filtering can produce unwanted results, as a shifted and attenuation signal relative to the original. Moreover, FDCNN allows overcoming the parameterization problem in nonlinear systems, which is often difficult to achieve. In order to validate the proposed methodology, a comparison between two different architectures of convolutional neural networks is made, showing that proposed CNN in frequency domain brings better performance in the identification system for damage diagnosis in building structures. Experimental results from reducing scale two-storey building confirm the effectiveness of the proposed.

**Keywords** Structural health monitoring · Damage detection · Convolutional neural network · Identification system

## 1 Introduction

Structural health monitoring (SHM) of a building structure during the lifetime, or after a seismic activity, has gained increasing attention in the engineering field. Over the past decades, vibration-based damage detection techniques have been studied extensively and have found increasing applications in civil and mechanical engineering. Most dam-

age detection methods typically involve data processing to explore changes in both the dynamic properties of structures such as vibration frequency (Farrar et al. 2001; Roux et al. 2014; Vidal et al. 2014) and mode shapes (Maia et al. 2003; Zhu et al. 2011; Rucevskis et al. 2016) directly related to the stiffness reduction as a consequence of structural damage. For practical applications, both methods require to excite a building at high frequencies, which is not easy to achieve, and therefore, the damage may go unnoticed. Following this line, the bulk of research (Doebbling et al. 1998; Zou et al. 2000; Carden and Fanning 2004; Fan and Qiao 2011; Pau and Vestroni 2013) provides an extensive summary review of vibration-based damage identification methods. The authors discuss the advantages and limitations of different methods under this approach. For a recent review based on vibration-based damage detection, readers can consult Das et al. 2016; Kong et al. 2017. The accuracy of those methods depends on many sensors, and they might be biased by measurement noise (Rahai et al. 2007).

Communicated by V. Loia.

✉ Jesús Morales-Valdez  
jmoralesv@ctrl.cinvestav.mx

Mario Lopez-Pacheco  
mlopezp@ctrl.cinvestav.mx

Wen Yu  
yuw@ctrl.cinvestav.mx

<sup>1</sup> Departamento de Control Automático, CINVESTAV-IPN, Mexico City, Mexico

<sup>2</sup> Cátedras CONACYT, Departamento de Control Automático, CINVESTAV-IPN, Mexico City, Mexico

Other studies as in Loh et al. (2011) suggest that during earthquakes, buildings structures exhibit a nonlinear and hysteretic behavior. Within this context, in Farrar et al. (2007) authors point out that under cyclic excitation associated with earthquakes, degradation of a structure manifests itself in the evolution of the associated hysteresis loop. The idea behind is that the plastic strain amplitude is related to the number of cycles to failure, and it can be represented by means of stress strain loops (Ma et al. 2004). Similarly, in Ikhouane et al. (2005) it is reported that structural damage caused by earthquake may be due to excessive deformations, or it may be in the form of accumulated damage sustained under repeated load reversals. Under this line, reference (Chatzi et al. 2010) presents a review of some damage detection methods, where damage-sensitive data features are based on nonlinear system response. In Ceravolo et al. (2013), the Bouc–Wen hysteretic model is employed to identify physical parameters such as stiffness degradation, strength deterioration and hysteresis behavior of the reinforced concrete frame, to be used as a safety assessment index for the seismic assessment of RC building. Moreover, an extensive bulk of research of vibration-based nonlinear system identification for damage detection can be found in Bursi et al. (2013). Applications of Bouc–Wen model and online identification for full three-dimensional scale steel-concrete can be found in Shan et al. (2016) and Wan et al. (2018). It is important to note that when the nonlinearity is known, nonlinear identification may reasonably follow a parametric approach and estimates response matches experimental data. Otherwise, the estimation is not always achieved and it is not possible to guarantee parametric convergence, due to measurement noise, offset and uncertainty. The most usual form to avoid measurement noise and offset is by using band pass filter. However, a prior system bandwidth is required to obtain satisfactory results; otherwise, correct filtering of the signal cannot be guaranteed, producing unwanted estimated values.

With respect uncertainty during experiments, data collection, measurement process or when determining the initial values. Arqub et al. (2016) proposes a new method to solve numerically fuzzy differential equations based on the use of the reproducing kernel as a potential tool to model several real physical phenomena under possibility uncertainty. This method yields more accurate approximations, especially in nonlinear cases. In the same research direction, a new efficient iterative algorithm for solving the analytic and approximate solutions of second-order, two-point fuzzy boundary value problems by using the Reproducing Kernel Hilbert Space method under the assumption of strongly generalized differentiability is investigated in Arqub et al. (2017). Similarly, in Arqub and Abo-Hammour (2014) the employment of continuous genetic algorithms is proposed in order to numerically approximate a solution of linear and nonlinear systems of second-order boundary value problems.

Reported results showed that the three methods are fast, accurate and very effective with a great potential in mathematical and engineering applications. However, in both cases, methods have not been evaluated for damage detection task.

Regarding nonlinear system identification, the artificial neural network approach has been widely used in characterizing structure-unknown nonlinear systems. The neural network framework offers a rigorous basis for identification systems, mainly because this approach does not require a mathematical model of the system. Neural networks also overcome the problem of parameterization of nonlinear systems, and their structure can be modified for each case (Chen et al. 1990). An exhaustive review of these methods can be found in Sohn et al. (2003). However, its application to physical systems is not always robust and accurate, and some methods demand long-time histories from the undamaged structure and intensive data processing, which is not always easy to achieve. Recently, the rapid advances in computation power has led to the use of deep learning techniques like convolutional neural networks (CNN) (LeCun et al. 1989) as a promising tool. The difference between the classical neural network (NN) and CNN are

- (1) CNN includes at least a convolutional layer, where units are not connected to all units in previous layer as in a fully connected layer and they are only connected to units near to them; it can be said that their receptive field is small.
- (2) The filters of CNN are shared in the same convolution layers. This allows to reduce the parameters to be trained, and certain properties or features from input data can be detected no matter its locations in the data.
- (3) The main layers of CNN are the subsample layers or pooling layers. These layers reduce the sizes of the data through the neural network, allowing deeper structures cause these layers do not require updatable parameters. This characteristic is especially beneficial because they reduce the number of units in the neural network.

Therefore, CNN is getting popular, especially for image classification with broad usage taking relevance in fields such as the automotive sector, industries, medicine, robotics and others. Satisfactory results under this approach are reported in Kim (2014) and Simard et al. (2003) that use CNN for sentence classification and document analysis, face recognition (Lawrence et al. 1997), road sign detection and classification (Bouti et al. 2018), Chinese license plates recognition (Liu et al. 2018b), bearing defects classification (Appana et al. 2018) and ImageNet LSVRC-2010 contest (Krizhevsky et al. 2012). Moreover, positive results have also been reported for detection task by using CNN. An accurate lithography hotspot detection framework is addressed by a CNN in reference (Shin and Lee 2016), obtaining better results and higher

performance in the ICCCAD 2012 dataset, also achieving a time reduction compare to optical simulation methods and SVM. Within some new CNN applications, we have an automated identification of abnormal EEG signals (Yıldırım et al. 2018), Alzheimer's disease detected by using magnetic resonance images and CNN (Vu et al. 2018), real-time ozone concentration prediction system (Eslami et al. 2019) and application on rotating machines failure detection (Udmale et al. 2019; Ma et al. 2019). Another recent application of CNN arisen in recent years is in civil engineering. CNN applied to buildings is very sensitive to damage assessment, because CNN takes advantage of minimal engagement of signal processing and automated features extraction for the fault diagnosis. In Cha et al. (2017), the convolutional neural network is used as a classifier over concrete cracks images to determine damage. In Lin et al. (2017), a convolutional neural network is used as a classifier for damage detection from data obtained from a low-level sensor and for feature detection at the same time. One-dimensional CNN for real-time damage detection is proposed in Abdeljaber et al. (2017); they propose to use a CNN for each joint and this way they can determine whether there is damage and locate it in a fast fashion. Following this line, Modarres et al. (2018) proposes a convolutional neural network basis on a computer vision approach in automated inspection to identify the presence and type of structural damage. Conducted simply from images Atha and Jahanshahi (2018), evaluates corrosion assessment on metallic surfaces using a different convolutional neural network and images. Other applications and variations of CNN to other specific damage detection contexts can be found in Zhao et al. (2019) and Liu et al. (2018a). In the latter, several algorithms with applications in rotary machines are presented. Note that most methods for damage diagnosis under CNN approach have reported satisfactory results in the analysis of images, and generally, they are developed on time domain. Since CNN incorporates random filters in its design, it reduces measurement noise.

However, the convolution in the time domain is increment operation that can require higher computation time with respect to other algorithms. Moreover, supposing that random filters do not completely eliminate measurement and offset, estimation can be biased of real data. An alternative to avoid these problems is to introduce frequency domain CNN (FDCNN) that adds an spectral pooling layer to reduce the measurement noise. The detailed reasons for using frequency domain CNN to estimate the hysteretic displacement are:

- (1) CNN is getting popular for image classification with a broad usage spanning across automotive, industrial, medicine, robotics and others. The convolution operation is change for a element-wise product that reduces the operations amount. This advantage is reflected on the training stage because the algorithm requires to realize

on each iteration many of these operations. CNN takes advantage of minimal engagement of signal processing and automated features extraction for the fault diagnosis. CNN applied to buildings is very sensitive to damage assessment.

- (2) FDCNN avoids memory size growth compared to traditional CNN based approach. FDCNN avoid the convolution stage.
- (3) FDCNN does not require any assumption on the type and localization of structural nonlinearity.
- (4) FDCNN does not require preprocessing stage and automatically learns directly from the vibration data and eliminates the noise components of the signal augmenting the system response that makes it robust to identification task.

In the past, several new research projects have been funded to improve the damage detection methods, including the use of innovative signal processing, new sensors and control theory. This paper highlights these new research directions and uses FDCNN to learn features directly from frequency data of vibration signals for damage detection in a building structure. The damage detection method is based on dissipated energy. Since the earthquake introduces several stress cycles in different directions in the structure, load-strain curves can be used as an indicator of damage. To represent these phenomena, a Bouc–Wen model is used, which is estimated through frequency domain CNN. It has been previously described that CNN has an outstanding performance as a classifier, but in the authors knowledge, there are not reported works that show FDCNN is used for system identification. The objects of the paper are:

- (1) We use the frequency domain CNN to model the hysteretic displacement via vibration data. Then, we apply the hysteretic displacement for the damage diagnosis.
- (2) Since the measurement noise and offset affect the identification systems, we use FDCNN to overcome it . The combination of frequency random filters and spectral pooling avoids measurement noise effect in the identification process. Therefore, the robustness of proposed algorithm is evaluated.

The main result in this paper is to show that the properties of FDCNN have some advantages over the time domain ones and NN when measurement noise in data exists as follows:

- (1) FDCNN overcome nonlinear parameterization in the identification system that is generally difficult to achieve. It can extract most important damage-sensitive characteristics automatically from acceleration signals.
- (2) The proposed algorithm is alternative solutions to the identification methods, which is robust to high-

frequency measurement noise. In the frequency domain, the convolution stage is replaced by elements-wise product that reduces computational complexity, as well as the execution time.

- (3) The proposed method avoids long-time histories from the undamaged structure and intensive data processing. Moreover, it can work at both large and small scales, depending on the number and location of sensors. In our paper, the intermediate scale approach is taken focusing on the detection of damage at a storey level.

The structure of the paper is the following: First, the mathematical model of a building structure and the Bouc–Wen hysteretic model are presented in Sect. 2. The architecture of proposed frequency domain convolutional neural network (FDCNN) for system identification task is described in Sect. refSec:CNNf, as well as a frequency analysis in a convolutional layer and a sensibility analysis of FDCNN to noise data. Section 4 contains the experimental results conducted in a reduced scale two-storey building prototype in order to investigate the damage detection capability. Moreover, a comparison study between neural network (see “Appendix C”), time domain CNN (see “Appendix B”) and proposed FDCNN is carried out to evaluate the performance of proposed method. Finally, a summary of the findings is provided in Sect. 5.

## 2 Mathematical model of building structure

The dynamics of a multiple degrees of freedom (MDOF) shear building structures subject to seismic activity is described by

$$M\ddot{x}(t) + C\dot{x}(t) + \mathcal{K}x(t) = -Ml\ddot{x}_g(t) \tag{1}$$

where

$$x(t) = \{x_1(t), x_2(t), \dots, x_n(t)\}^T \in \mathfrak{R}^{n \times 1}, \tag{2}$$

$$\dot{x}(t) = \{\dot{x}_1(t), \dot{x}_2(t), \dots, \dot{x}_n(t)\}^T \in \mathfrak{R}^{n \times 1}, \tag{3}$$

$$\ddot{x}(t) = \{\ddot{x}_1(t), \ddot{x}_2(t), \dots, \ddot{x}_n(t)\}^T \in \mathfrak{R}^{n \times 1}, \tag{4}$$

$$l = \{1, 1, \dots, 1\}^T \in \mathfrak{R}^{n \times 1}, \tag{5}$$

$$\ddot{x}_a(t) = \ddot{x}(t) + l\ddot{x}_g(t) \in \mathfrak{R}^{1 \times 1} \tag{6}$$

The term  $n$  indicates the number of floors; the entries  $x_i(t)$ ,  $\dot{x}_i(t)$  and  $\ddot{x}_i(t)$ , with  $i = 1, 2, \dots, n$ , are the relative displacement, velocity and acceleration of each floor, respectively, measured with respect to the basement. Signal  $\ddot{x}_a(t)$  represents the absolute acceleration, and  $\ddot{x}_g(t)$  is the ground acceleration induced by the earthquake that is distributed by the influence vector  $l$ . Moreover,  $M$ ,  $\mathcal{K}$  and  $C$  are the mass,

stiffness and damping matrices, respectively, defined as

$$M = \begin{bmatrix} m_1 & 0 & \dots & 0 \\ 0 & m_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & m_n \end{bmatrix} > 0 \in \mathfrak{R}^{n \times n} \tag{7}$$

$$C = \begin{bmatrix} c_1 + c_2 & -c_2 & \dots & 0 \\ -c_2 & c_2 + c_3 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & c_n \end{bmatrix} \geq 0 \in \mathfrak{R}^{n \times n} \tag{8}$$

$$\mathcal{K} = \begin{bmatrix} \kappa_1 + \kappa_2 & -\kappa_2 & \dots & 0 \\ -\kappa_2 & \kappa_2 + \kappa_3 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \kappa_n \end{bmatrix} > 0 \in \mathfrak{R}^{n \times n} \tag{9}$$

where parameters  $c_i$  and  $\kappa_i$  are, respectively, the lateral column damping and stiffness between the  $i$ th and  $(i - 1)$ th storey.

Note that the damping at the building structure is represented by Rayleigh model, Chopra (1995) defined by

$$C = a_0M + a_1\mathcal{K} \tag{10}$$

where the Rayleigh parameters  $a_0$  and  $a_1$  are calculated by using the first and third eigen-frequencies  $\omega_i$ , in the following expression

$$\frac{1}{2} \begin{bmatrix} \frac{1}{\omega_i} & \omega_i \\ \frac{1}{\omega_j} & \omega_j \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \xi_i \\ \xi_j \end{bmatrix} \tag{11}$$

where  $\xi_j$   $\omega_i$  with  $i = j = 1, 2, \dots, n$  are the damping ratio and the vibration frequency of the  $i$ th structural mode, respectively. Note that model (1) assumes that the building structure is undamaged and operates in its elastic range.

**Remark 1** Initially the building is at rest, that is,  $x(0) = \dot{x}(0) = \ddot{x}(0) = 0$ . Moreover, ground acceleration is zero before an earthquake  $\ddot{x}_g = 0$ .

**Remark 2** Acceleration measurements of each storey and basement are available, and they are affected by offset and high-frequency measurement noise.

$$\ddot{x}_m = \ddot{x} + \varsigma + \lambda \tag{12}$$

$$\ddot{x}_{gm} = \ddot{x}_g + \varsigma_g + \lambda_g \tag{13}$$

with  $\ddot{x}_m = [\ddot{x}_{1m} \ \ddot{x}_{2m} \ \dots \ \ddot{x}_{nm}]$  is the measured acceleration vector,  $\ddot{x}_{gm}$  is the measured ground acceleration,  $\varsigma = [\varsigma_1 \ \varsigma_2 \ \dots \ \varsigma_n]$  and  $\varsigma_g$  are measurement offsets, and  $\lambda = [\lambda_1 \ \lambda_2 \ \dots \ \lambda_n]$  and  $\lambda_g$  are high-frequency measurement noises. For ease  $\ddot{x}_m$  will be considered as  $\ddot{x}$  throughout the article.

**Remark 3** Assuming that the building is damaged, a nonlinear degradation term is introduced in (1) that relates the strain-stress with damage

$$M(\ddot{x} + l\ddot{x}_g) + C\dot{x} + T\rho(x, z) = 0 \tag{14}$$

$$T = \text{diag} [1, 1, \dots, 1] \tag{15}$$

$$\rho(x, z) = [\rho(x_1, z_1), \rho(x_2, z_2), \dots, \rho(x_n, z_n)]^T \tag{16}$$

where nonlinearity  $\rho(x, z)$  is represented by using the smooth hysteresis Bouc–Wen model, Wen (1976)

$$\rho(x_i, z_i) = \alpha_i \kappa_i x_i + (1 - \alpha_i) \kappa_i z_i \tag{17}$$

$$\dot{z}_i = \frac{A_i \dot{x}_i - v_i (\beta_i |\dot{x}_i| z_i^{\sigma_i - 1} z - \gamma_i \dot{x}_i |z_i|^{\sigma_i})}{\eta_i} \tag{18}$$

where the subscript  $i = 1, 2, \dots, n$  refers to the floor number;  $\alpha, \kappa$  and  $\gamma$ , are the ratio of postyield, the preyield stiffness and the yield deformation, respectively, whereas  $z_i$  is the hysteretic displacement of the nonlinear shear building. Generally,  $\beta$  and  $\gamma$  are called loop parameters and they affect the size, whereas  $\sigma > 1$  influences the smoothness on the hysteresis loop. Moreover,  $v$  and  $\eta$ , are strength and stiffness degradation functions of the dissipated hysteretic energy, respectively, defined as (Ma et al. 2006),

$$\eta_i(E_i) = 1.0 + \delta_{\eta,i} E_i \tag{19}$$

$$v_i(E_i) = 1.0 + \delta_{v,i} E_i \tag{20}$$

where  $\delta_\eta$  and  $\delta_v$  are the stiffness and strength degradation ratio, respectively. Generally, these variables are nonnegative and unknown parameters, that will be estimated.

**Remark 4** A convenient measure of degradation as a result of structural damage is the dissipated energy from structural hysteresis cycle measured from  $t = 0$  to  $t$

$$E_i(t) = \int_0^t z_i \dot{x}_i dx \tag{21}$$

Note that the systems described in (14) and (18) can be rewritten as a set of nonlinear differential equations subjected to the external force

$$m_1 \ddot{x}_1 + c_1 \dot{x}_1 + \rho(x_1, z_1) = m_1 \ddot{x}_g \tag{22}$$

$$m_2 \ddot{x}_2 + c_2 \dot{x}_2 + \rho(x_2, z_2) = m_2 \ddot{x}_g \tag{23}$$

⋮

$$m_n \ddot{x}_n + c_n \dot{x}_n + \rho(x_n, z_n) = m_n \ddot{x}_g \tag{24}$$

equivalents to

$$m_1 \ddot{x}_1 + c_1 \dot{x}_1 + \alpha_1 \kappa_1 x_1 + (1 - \alpha_1) \kappa_1 z_1 = m_1 \ddot{x}_g \tag{25}$$

$$m_2 \ddot{x}_2 + c_2 \dot{x}_2 + \alpha_2 \kappa_2 x_2 + (1 - \alpha_2) \kappa_2 z_2 = m_2 \ddot{x}_g \tag{26}$$

⋮

$$m_n \ddot{x}_n + c_n \dot{x}_n + \alpha_n \kappa_n x_n + (1 - \alpha_n) \kappa_n z_n = m_n \ddot{x}_g \tag{27}$$

Taking into account that parameters and the internal state  $z_i$  of Bouc–Wen hysteretic model (18) are unknown, then both must be estimated, as

$$\dot{\hat{z}}_i = \frac{\hat{A}_i \dot{x}_i - \hat{v}_i (\hat{\beta}_i |\dot{x}_i| \hat{z}_i^{\sigma_i - 1} \hat{z} - \hat{\gamma}_i \dot{x}_i |\hat{z}_i|^{\sigma_i})}{\hat{\eta}_i} \tag{28}$$

$$\hat{\eta}_i(E_i) = 1.0 + \delta_{\eta,i} \hat{E}_i \tag{29}$$

$$\hat{v}_i(E_i) = 1.0 + \delta_{v,i} \hat{E}_i \tag{30}$$

$$\hat{E}_i(t) = \int_0^t \hat{z}_i \dot{x}_i dx \tag{31}$$

In this work, FDCNN is proposed to identify the Bouc–Wen hysteretic displacement 28, as an important application for damage detection in building structures through an energy analysis. The use of CNN in real application overcomes the parameter and state estimation problem. The inclusion of random filters in the FDCNN design eliminates measurement noise in acceleration data, as will be shown later.

### 3 Frequency domain CNN architecture

In this section, the development of the proposal frequency domain CNN is presented. The main differences with respect time domain CNN is that a discrete Fourier Transform (DFT) is applied to the inputs and to the filters in convolutional layers; thus, the operations become simpler from the computational point of view. Also no activation function is required. Definition and uses of DFT in the FDCNN are shown in “Appendix A.”

Consider an unknown discrete-time nonlinear system

$$y(q) = f(x(q)). \quad x(q + 1) = g(x(q), u(q)) \tag{32}$$

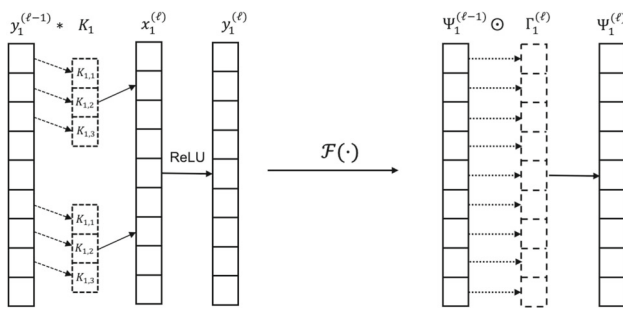
where  $y(q)$  is the scalar output,  $x(q)$  the internal state,  $u(q)$  the input,  $f(\cdot)$  and  $g(\cdot)$  smooth functions,  $f, g \in C^\infty$ .

A nonlinear autoregressive exogenous (NARX) model for (32) is defined as

$$y(q) = \Phi[\varpi(q)] \tag{33}$$

and the system dynamics are represented by the unknown nonlinear difference equation  $\Phi$ , where

$$\varpi(q) = [y(q - 1), \dots, y(q - n_y), u(q), \dots, u(q - n_u)]^T \tag{34}$$



**Fig. 1** Convolutional layer operations

$y(q)$  and  $u(q)$  into (34) represent, respectively, the measurable output and input for the system, with  $n_y$  and  $n_u$  the regression order, respectively, which are unknown.

Consider the system (33) to be estimated and regard the same input for the CNN. A discrete Fourier transform (DFT) is applied to this input to obtain a frequency representation of the same length, i.e.,  $\Phi^{(0)} = \mathcal{F}(\hat{w})$ . In this representation, it is assumed that DC frequencies are shifted to the center of the domain.

In (35), the output layer of this new frequency domain convolutional neural network (DFCNN) is shown, where  $\hat{y}_F(q)$  is the scalar output signal of the FDCNN. This layer now is a fully connected layer, with  $\Upsilon$  is the output of the last subsample layer and  $V^{(\ell)} \in R^{L_2}$  are the weights in the output layer.

$$\hat{y}_F(q) = V^{(\ell)T} \Upsilon \tag{35}$$

For the convolutional layers, random filters are defined like  $q_i^{(\ell)} \in \mathbb{R}^{f_\ell}$ , where  $i = 1, 2, \dots, h_2$ . Furthermore,  $h_2$  is the total of filters in the current layer  $\ell$ . These filters also go through a DFT to match the dimension of the output of the previous layer (for the first layer, the transform will match the size of  $\Phi^{(0)}$ ), i.e.,  $\Gamma_i^{(\ell)} = \mathcal{F}(q_i^{(\ell)})$ , for this conversion, matrix  $F$  defined in section A, a new matrix is built for each size of the data. So, the output of a convolutional layer is defined as the element-wise product ( $\odot$ ) of the output of previous layer and the filters, such as

$$\Psi_i^{(\ell)} = \Psi_i^{(\ell-1)} \odot \Gamma_i^{(\ell)} \tag{36}$$

Figure 1 shows how both convolutional layers work. While in time domain filters has less elements than the filters in frequency domain, the convolution between them and the input requires more operations. Also no activation function is used in frequency domain as it can be seen.

For the subsample layers, a spectral pooling operation is applied (Rippel et al. 2015). Here, the idea is to remove high frequencies to reduce size of the input.  $s^{(\ell)}$  represents the number of elements to be removed, so output of these layers

**Table 1** Algorithm 1: spectral pooling

Algorithm 1:	Spectral pooling
<b>Input:</b>	Map $x \in \mathbb{R}^{M \times N}$ , output size $H \times W$
<b>Output:</b>	Pooled map $\hat{x} \in \mathbb{R}^{H \times W}$
1:	$y \leftarrow \mathcal{F}(x)$
2:	$\hat{y} \leftarrow \text{CROPSPECTRUM}(y, H \times W)$
3:	$\hat{y} \leftarrow \text{TREATCORNERCASE}(\hat{y})$
4:	$\hat{x} \leftarrow \mathcal{F}^{-1}(\hat{y})$

**Table 2** Algorithm 2: spectral pooling proposed

Algorithm 2:	Spectral pooling
<b>Input:</b>	Map $\Psi_i^{(\ell-1)} \in \mathbb{C}^N$ , output size $N - s^{(\ell)}$
<b>Output:</b>	Pooled map $\Psi_i^{(\ell)} \in \mathbb{C}^{N-s^{(\ell)}}$
1:	$\Psi_i^{(\ell)} \leftarrow \text{SHRINK}(\Psi_i^{(\ell-1)}, s^{(\ell)})$

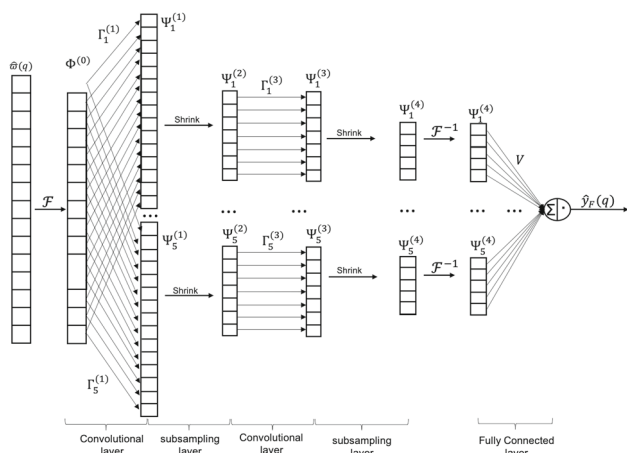
is defined as

$$\Psi_i^{(\ell)} = \text{SHRINK}(\Psi_i^{(\ell-1)}, s^{(\ell)}) \tag{37}$$

The Shrink operation in the spectral pooling removes  $s^{(\ell)}$  elements in its input, two from the top and two from its bottom, so the output remains symmetric. Initially, this operation was introduced by Rippel et al. (2015). The Shrink is defined in Table 1:

For subsample layer, the *shrink* operation is based on algorithm presented in Table 2. Since Algorithm 1 is intended for a general case where matrices are used, some modifications were made. The first modification consists in eliminating the two steps where the DFT operation is performed as well as its inverse. Given the structure of FDCNN, it is not necessary to be passing between domains (time to frequency and vice versa) in each layer of the network, but only in the input layer and at the end of the subsampling layers. The second modification is made by eliminating step 3 of the algorithm, because this operation deals with the case where the representations do not have the appropriate dimension and therefore a real output cannot be obtained. This problem is eliminated by proposing an input with adequate dimensions in the network, in such a way that the convolutional and subsampling layers and the inverse DFT are applied; hence, real values are obtained. Finally, given that the proposed FDCNN works with vectors instead of matrices, step 2 is carried out while retaining the central sub vector of the input. These modifications are reflected in Table 2.

Note that all representations in frequency domain have an odd size dimension to simplify calculations and in further operations we can obtain a time domain representation adequate. As previously mentioned, as many as they are needed, convolutional and subsample layers can be connected one



**Fig. 2** Frequency domain convolutional neural network for system identification

after other. After this cascade connection of layers, the output of the last subsample layer has to be mapped back, i.e.,  $\psi_i^{(\ell)} = \mathcal{F}^{-1}(\Psi_i^{(\ell)})$  and stacked in a single vector

$$\mathcal{Y} = [\psi_1^{(\ell)T}, \psi_2^{(\ell)T}, \dots, \psi_{h_2}^{(\ell)T}]^T \tag{38}$$

The complete architecture of FDCNN is illustrated in Fig. 2. The equations for training are described in the next section.

### 3.1 Training of frequency domain CNN

The backpropagation algorithm is used for training. The cost function is

$$J(q) = \frac{1}{2} e^2(q) = [\hat{y}_F(q) - y(q)]^2 \tag{39}$$

For the fully connected layer, the gradient of  $J$  with respect to synaptic weights  $V$  is:

$$V(q + 1) = V(q) - \eta_o \frac{\partial J}{\partial V} = V(q) - \eta_F e \mathcal{Y} \tag{40}$$

$\eta_F$  is the learning rate defined one for each layer.

The propagated error to the previous layer is

$$\frac{\partial J}{\partial \mathcal{Y}} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial \hat{y}_F} \frac{\partial \hat{y}_F}{\partial \mathcal{Y}} = e V \tag{41}$$

Because  $\mathcal{Y}$  is the stacked vector of the inputs of subsample layers, we take the same amount of elements that each  $\psi_1^{(\ell)}$  gave in forward stage. Next, the propagated gradient to each one of this outputs has to be transformed using the DFT by applying the DFT matrix inverse corresponding to the size we want to match, so it can be propagated through the subsample and convolutional layers.

For the subsample layers, the spectrum has to equal the size of the previous convolutional layer, so, in this layer the only operation required is to increase the frequency representation of the propagated error, i.e.,

$$\frac{\partial J}{\partial \Psi_i^{(\ell-1)}} = up \left( \frac{\partial J}{\partial \Psi_i^{(\ell)}} \right) \tag{42}$$

where  $up(\cdot)$  is an operation realized to increase the spectral representation. In order to realize this operation after the gradient goes through a convolutional layer, which is already in frequency domain, properties of DFT are used and a series of matrix multiplication are realized to match the size of previous layers; this prevents in some way making the inverse transform to time domain and back again to frequency domain.

For the convolutional layers, the update is as follows:

$$\frac{\partial J}{\partial \Gamma^{(\ell)}} = \frac{\partial J}{\partial \Psi^{(\ell)}} \odot \Psi^{(\ell-1)} \tag{43}$$

This is the element-wise product between the propagated error and the output of previous layer. Since each element of the filter can be updated separately, then (43) can be written as:

$$\frac{\partial J}{\partial \Gamma_a^{(\ell)}} = \frac{\partial J}{\partial \Psi_a^{(\ell)}} \Psi_a^{(\ell-1)} \tag{44}$$

$\Gamma_a^{(\ell)}$  is each element of the filters represented in frequency domain with  $a = 1, 2, \dots, f_\ell$ .

To obtain the propagated gradient to the previous layers, we have

$$\frac{\partial J}{\partial \Psi^{(\ell-1)}} = \frac{\partial J}{\partial \Psi^{(\ell)}} \odot \Gamma^{(\ell)} \tag{45}$$

which is also the element-wise product between the filters and the propagated gradient to the current layer.

### 3.2 Frequency analysis in a convolutional layer

The convolutional layer as defined in (36) represents the element-wise product between the output of previous layer and the filters in the current layer, and it is interesting to obtain some more information about this operation; in that sense, a proposition is formulated showing the relationship between the output changes with respect to the input change into a convolutional layer.

**Proposition 1** Consider a convolutional layer in a FDCNN defined as:

$$\Psi_i^{(\ell)} = \Psi_i^{(\ell-1)} \odot \Gamma_i^{(\ell)} \tag{46}$$

where  $\Psi_i^{(\ell)}$  is the output of the current layer  $\ell$ ,  $\Psi_i^{(\ell-1)}$  represent the output of previous layer, in the case  $\ell = 1$ ,  $\Psi_i^{(0)} = \Phi^{(0)}$ ,  $\Gamma_i^{(\ell)}$  is the frequency domain representation of the filters in this layer and  $i = 1, 2, \dots, h$ , where  $h$  is total number of hyperparameters in the layer.

The relationship between  $\Delta\Psi_i^{(\ell)}(q)$  and  $\Delta\Psi_i^{(\ell-1)}(q)$  is

$$\frac{\Delta\Psi_i^{(\ell)}(q)}{\Delta\Psi_i^{(\ell-1)}(q)} = \Gamma_i^{(\ell)}(q) \quad (47)$$

when the filters are updated using the following rule

$$\Delta\Gamma_i^{(\ell)}(q) = \frac{\Psi_i^{(\ell)}(q-1) - \Gamma_i^{(\ell)}(q)\Psi_i^{(\ell-1)}(q-1)}{\Delta\Psi_i^{(\ell-1)}(q)} \quad (48)$$

**Proof** Differentiating (46), it is obtained

$$\Delta\Psi_i^{(\ell)}(q) = \Delta\Gamma_i^{(\ell)}(q)\Psi_i^{(\ell-1)}(q) + \Gamma_i^{(\ell)}(q)\Delta\Psi_i^{(\ell-1)}(q) \quad (49)$$

with the aid of the definitions  $\Delta\Psi_i^{(\ell-1)}(q) = \Psi_i^{(\ell-1)}(q) - \Psi_i^{(\ell-1)}(q-1)$ ,  $\Delta\Psi_i^{(\ell)}(q) = \Psi_i^{(\ell)}(q) - \Psi_i^{(\ell)}(q-1)$  and  $\Delta\Gamma_i^{(\ell)}(q) = \Gamma_i^{(\ell)}(q) - \Gamma_i^{(\ell)}(q-1)$ , it is followed that

$$\begin{aligned} \Delta\Psi_i^{(\ell)}(q) &= \Delta\Gamma_i^{(\ell)}(q)(\Psi_i^{(\ell-1)}(q) + \Psi_i^{(\ell-1)}(q-1)) \\ &\quad + \Gamma_i^{(\ell)}(q)\Delta\Psi_i^{(\ell-1)}(q) \\ &= \Delta\Gamma_i^{(\ell)}(q)\Psi_i^{(\ell-1)}(q-1) \\ &\quad + (\Delta\Gamma_i^{(\ell)}(q) + \Gamma_i^{(\ell)}(q))\Delta\Psi_i^{(\ell-1)}(q) \\ &= (\Gamma_i^{(\ell)}(q) - \Gamma_i^{(\ell)}(q-1))\Psi_i^{(\ell-1)}(q-1) \\ &\quad + (\Delta\Gamma_i^{(\ell)}(q) + \Gamma_i^{(\ell)}(q))\Delta\Psi_i^{(\ell-1)}(q) \\ &= \Gamma_i^{(\ell)}(q)\Psi_i^{(\ell-1)}(q-1) - \Psi_i^{(\ell)}(q-1) \\ &\quad + (\Delta\Gamma_i^{(\ell)}(q) + \Gamma_i^{(\ell)}(q))\Delta\Psi_i^{(\ell-1)}(q) \\ &= \Gamma_i^{(\ell)}(q)\Psi_i^{(\ell-1)}(q-1) - \Psi_i^{(\ell)}(q-1) \\ &\quad + \Delta\Gamma_i^{(\ell)}(q)\Delta\Psi_i^{(\ell-1)}(q) + \Gamma_i^{(\ell)}(q)\Delta\Psi_i^{(\ell-1)}(q) \end{aligned} \quad (50)$$

Defining

$$\Delta\Gamma_i^{(\ell)}(q) = \frac{\Psi_i^{(\ell)}(q-1) - \Gamma_i^{(\ell)}(q)\Psi_i^{(\ell-1)}(q-1)}{\Delta\Psi_i^{(\ell-1)}(q)} \quad (51)$$

replacing (51) in (50)

$$\Delta\Psi_i^{(\ell)}(q) = \Gamma_i^{(\ell)}(q)\Delta\Psi_i^{(\ell-1)}(q) \quad (52)$$

or

$$\frac{\Delta\Psi_i^{(\ell)}(q)}{\Delta\Psi_i^{(\ell-1)}(q)} = \Gamma_i^{(\ell)}(q) \quad (53)$$

**Remark 5** Proposition 1 shows that using a different training rule over the filters of a convolutional layer allows to reach a proportional relationship between the output variations and the input variations.

**Remark 6** This relationship can be applied for a direct analysis on several convolutional layers connected in cascade, showing that variations in the input data are only affected proportionally, decreasing or increasing the main components of these data according to each filter.

### 3.2.1 Sensibility of FDCNN to noisy data

Proposition 1 shows the relationships in a convolutional layer, and this result can be extended to a cascade connection of convolutional layer, using a ReLU activation function after each operation and spectral pooling layer, it yields

$$\Delta\Psi_i^{(\ell)} = SP\left(f\left(\Gamma_i^{(\ell)} \odot \dots \odot SP\left(f\left(\Gamma_i^{(2)} \odot SP\left(f\left(\Gamma_i^{(1)} \odot \Delta\Phi^{(0)}\right)\right)\right)\right)\right)\right) \quad (54)$$

For ease notation, the instant ( $q$ ) is omitted, but this analysis can be done in each iteration of the FDCNN. The activation function  $f$  keeps the positive part of its argument; otherwise, set them to zero and they can be omitted. Hence, the analysis is carried out by means of positive values after the activation function. Therefore, Eq. (54) can be rewritten as

$$\Delta\Psi_i^{(\ell)} = SP\left(\Gamma_i^{(\ell)} \odot \dots \odot SP\left(\Gamma_i^{(2)} \odot SP\left(\Gamma_i^{(1)} \odot \Delta\Phi^{(0)}\right)\right)\right) \quad (55)$$

The spectral pooling operation reduces the frequency representation of its arguments by eliminating the highest frequency component and its conjugate. Thus, elements in (55) will not be considered for the analysis, reducing the expression to

$$\Delta\Psi_{i,a}^{(\ell)} = \Gamma_i^{(\ell)}\Gamma_{i,a}^{(\ell-1)} \dots \Gamma_{i,a}^{(1)}\Delta\Phi_a^{(0)} = \Gamma_T\Phi_a^{(0)} \quad (56)$$

where  $a = 1, 2, \dots, q$ ,  $q$  indicates the remaining elements that were not eliminated by the spectral pooling. This equation represents the interaction between the convolutional output, pooling layers and the FDCNN input. Considering the input defined in (57) that includes measurement noise, i.e.,  $\Phi_a^{(0)} = \Phi_{a,0}^{(0)} + \lambda$ , where  $\lambda$  is a high-frequency bounded



noise, it is also assumed that the main frequency of the system is much lower than that of the noise.

$$\hat{w}(q) = [\hat{y}(q-1), \dots, \hat{y}(q-r_1), u(q), \dots, u(q-r_2)]^T \tag{57}$$

with  $r_1$  and  $r_2$  being the regression order.  $r_1 \neq n_y$  and  $r_2 \neq n_u$ . Under this assumption, absolute value of (56) is obtained

$$|\Delta\psi_{i,a}^{(\ell)}| = |\Gamma_T| |\Phi_a^{(0)}| \tag{58}$$

substituting  $\Delta\Phi_a^{(0)} = \Delta\Phi_{a,0}^{(0)} + \Delta\lambda$

$$|\Delta\psi_{i,a}^{(\ell)}| = |\Gamma_T| |\Delta\Phi_{a,0}^{(0)} + \Delta\lambda| \tag{59}$$

using the triangle property,

$$|\Delta\psi_{i,a}^{(\ell)}| \leq |\Gamma_T| |\Delta\Phi_{a,0}^{(0)}| + |\Gamma_T| |\Delta\lambda| \tag{60}$$

finally considering  $|\Delta\lambda| \leq \mathcal{M}$ , with  $\mathcal{M} \in \Re, \mathcal{M} > 0$

$$|\Delta\psi_{i,a}^{(\ell)}| \leq |\Gamma_T| |\Delta\Phi_{a,0}^{(0)}| + |\Gamma_T| \mathcal{M} \tag{61}$$

From (61), the first term corresponds to the system response through the layers in the FDCNN, whereas the second one corresponds to the effect of noise across the network. Since the spectral pooling is used, the highest impact frequency components are eliminated, leaving only the low-frequency components whose contribution to the response is minimal. In this way, the first term provides the greatest response, while the output is bounded to a region very close to it.

The internal structure of  $\Gamma_T$  represents the element-wise product of filters in each layer that represent filters in frequency domain. Considering that

$$\Gamma_{i,a}^{(j)} = \Re|\Gamma_{i,a}^{(j)}| + i\Im|\Gamma_{i,a}^{(j)}| \tag{62}$$

for  $j = 1, 2, \dots, \ell$ . For two convolutional layers, the filters part in (61) can be expressed as

$$|\Gamma_T| = \left| \Gamma_{i,a}^{(2)} \Gamma_{i,a}^{(1)} \right| \tag{63}$$

Moreover, using (62) a more detailed expression for noise data is found, where the interaction between real and imaginary parts of filters is shown

$$\begin{aligned} |\Gamma_T| &= \left| \left( \Re(\Gamma_{i,a}^{(2)}) + i\Im(\Gamma_{i,a}^{(2)}) \right) \left( \Re(\Gamma_{i,a}^{(1)}) + i\Im(\Gamma_{i,a}^{(1)}) \right) \right| \\ &= \left| \left( \Re(\Gamma_{i,a}^{(2)})\Re(\Gamma_{i,a}^{(1)}) - \Im(\Gamma_{i,a}^{(2)})\Im(\Gamma_{i,a}^{(1)}) \right) \right. \\ &\quad \left. + i \left( \Re(\Gamma_{i,a}^{(1)})\Im(\Gamma_{i,a}^{(1)}) + \Im(\Gamma_{i,a}^{(2)})\Re(\Gamma_{i,a}^{(1)}) \right) \right| \\ &= \left\{ \left( \Re(\Gamma_{i,a}^{(2)})\Re(\Gamma_{i,a}^{(1)}) - \Im(\Gamma_{i,a}^{(2)})\Im(\Gamma_{i,a}^{(1)}) \right)^2 \right. \\ &\quad \left. + \left( \Re(\Gamma_{i,a}^{(1)})\Im(\Gamma_{i,a}^{(1)}) + \Im(\Gamma_{i,a}^{(2)})\Re(\Gamma_{i,a}^{(1)}) \right)^2 \right\}^{1/2} \end{aligned} \tag{64}$$

In general, when adding more layers, the operations are repetitive and can be denoted as follows:

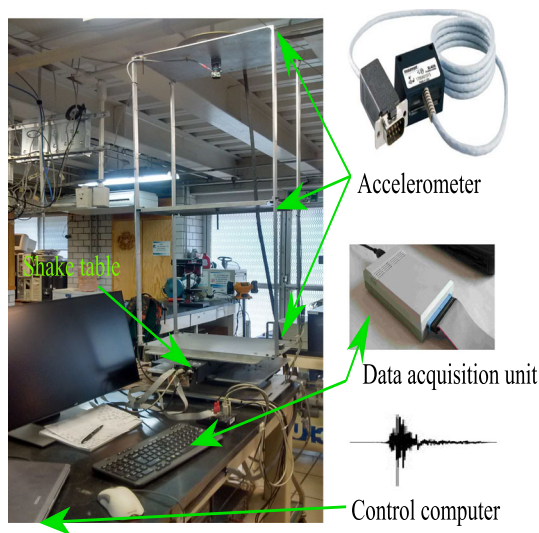
$$\begin{aligned} |\Gamma_T| &= \left| \prod_j^\ell \left( \Gamma_{i,a}^{(j)} \right) \right| \\ |\Gamma_T| &= \left| \prod_j^\ell \left( \Re(\Gamma_{i,a}^{(j)}) + i\Im(\Gamma_{i,a}^{(j)}) \right) \right| \\ |\Gamma_T| &= \sqrt{(\Re(\Gamma_T))^2 + (\Im(\Gamma_T))^2} \end{aligned} \tag{65}$$

The last equation shows the relationships between the real and imaginary parts of each filter and their interaction with others in different layers.

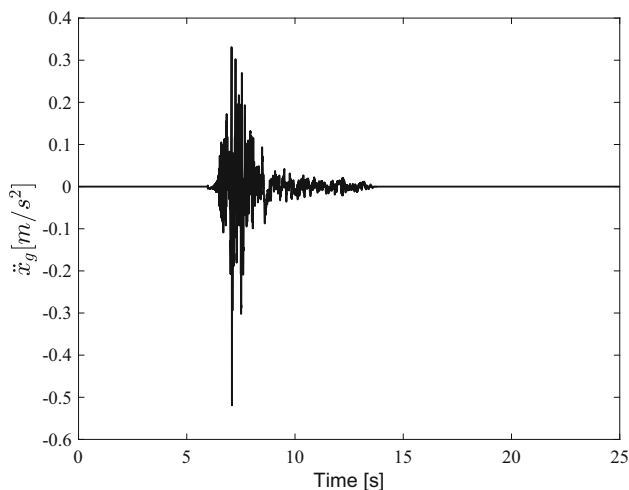
### 4 Experimental validation

The experimental two-storey building prototype used in this study is depicted in Fig. 3, constructed of aluminum with dimensions (32.5 × 53) cm and height of 1.2 m. All of columns have a rectangular cross section with width of (0.635 × 2.54) cm, with 58 cm of interstorey separation for the first floor and 62 cm for the remaining floor. The building is mounted over a shake table actuated by servomotors from Quanser, model I-40. During experiments, the structure is excited with the Northridge earthquake for a duration of 25 seconds that is fitted in amplitude to be in agreement with the structure and shown in Fig. 4. The building is equipped with Analog Devices accelerometers XL403A model, with a measuring range from 1 to 15 g and width band [1 × 800] Hz, to measure the responses at every storey and at the base. Data acquisition was carried out by using a RT-DAC/USB2 series electronic boards from Inteco. The acquisition programs were operated in Windows 7 with Matlab 2011a/Simulink. The communication between these boards and Simulink were carried out using C compiler.

From experiments, vibration frequencies of the reduce scale building structure are  $f_1 = 1.758$  Hz and  $f_2 = 4.0$  Hz, extracted by means of the Fourier spectra of the building



**Fig. 3** Experimental prototype



**Fig. 4** Northridge earthquake signal

acceleration data. On the other hand, from materials properties, preliminary information is obtained. Stiffness values  $k_1 = 12011 \text{ N/m}$  and  $k_2 = 12108 \text{ N/m}$  were calculated using the nominal values of the mechanical properties (Hibberler 2011), whereas the masses were measured directly giving  $m_1 = 2.034 \text{ kg}$  and  $m_2 = 2.534 \text{ kg}$ . Based on experimental data, damping Rayleigh is calculated assuming that the first two modes of the structure have a damping factor of 2%, i.e.,  $\xi_1 = \xi_2 = 0.02$ . The values were fixed, although during experiments  $\xi_1$  and  $\xi_2$  varied depending on the excitation signal. Moreover, assuming that during seismic activity only acceleration can be measured directly, velocity and displacement are estimates from available accelerations data  $\ddot{x}_i$ , with  $i = 1, 2, \dots, n$ . Estimates are obtained employing the following filter, consisting of two high-pass (hp) filters connected in cascade with the integrator, defined by

$$f(s) = \underbrace{\frac{s^2}{s^2 + 3.77s + 3.55}}_{hp} \times \underbrace{\frac{s^2}{s^2 + 3.77s + 3.55}}_{hp} \times \frac{1}{s} \quad (66)$$

where the cutting frequency is set at 0.3 Hz to remove the low-frequency components and to avoid drift.

For damage detection propose, the Bouc–Wen hysteretic model is introduced to represent the load deformation curves obtained during seismic activity test. The proposed method here postulated that stiffness loss reduces the capacity of the building to energy dissipation resulting from structural damage. In this sense, a system identification based on CNN is developed following the architecture shown in Fig. 5. From this, structural parameters are only employed to calculate analytical Bouc–Wen hysteretic state  $z_i$  required for CNN training stage in the identification system, by means of the backpropagation algorithm defined in previous sections. The hysteretic displacement signal corresponding to each storey is estimated by means of frequency domain CNN. Later, the FDCNN uses acceleration and velocity measured data for damage assessment, and analytical Bouc–Wen model is used as a reference signal to compare our results.

Despite the success reported in the literature about TDCNN, most applications to physical systems are mainly for image recognition. Unlike, this paper evaluates experimentally how the FDCNN performance can be affected when measured noise is presented in data. On the other hand, in most practical applications it is difficult to know accurately the structural-system bandwidth. So even, the implementation of signal preprocessing stages through filters also does not guarantee a good performance if the cutoff frequency does not match the system bandwidth. Therefore, finding the correct cutoff frequency can be a fairly difficult task to achieve. An alternative to these methods is to use FDCNN which incorporates random filters to strengthen the algorithm against measurement noise. In Sect. 3.2.1, a sensitivity analysis has been carried out that proves the sensitivity of FDCNN to overcome measurement noise. Moreover, the computing time is shorter compared to the TDCNN, as will be demonstrated in the experimental tests evaluated in the next section.

#### 4.1 System identification task

To validate the performance of frequency domain CNN, obtained results are compared with two different identification system scheme based on time domain convolutional neural network (TDCNN) and neural network (NN), respectively, described in “Appendices B and C.” All tests consist of vibration data containing measurement noise and offset. The final goal is to investigate the versatility to estimate the hysteretic state by using CNN and then make the damage diagnosis using the energy dissipated in the hysteretic cycle.

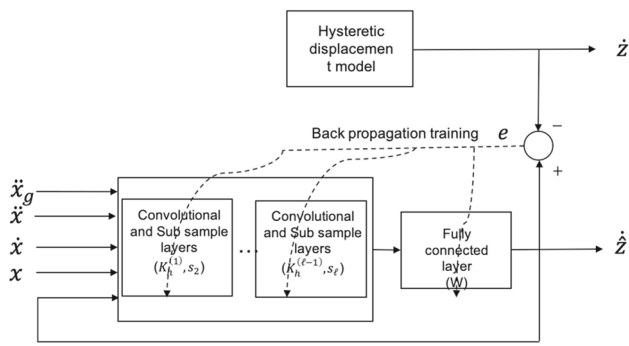


Fig. 5 System identification process

Experiments were carried out in a 2.6 GHz Intel Core i7 processor with 16 GB in RAM.

### 4.1.1 Identification system using frequency domain CNN

In this subsection, the frequency domain CNN is used to identify the hysteretic state  $z_i$  at each storey. It is important to note that the FDCNN results will be used as reference to be compared with results employing TDCNN and NN.

The proposed frequency domain convolutional neural network (FDCNN) consists of 2 convolutional layers  $Conv1f$  and  $Conv3f$  each one with 5 filters,  $h_2 = 5$ , the length of the filters are  $f_1 = f_3 = 3$ . Two subsample layers  $Sub2$  and  $Sub4$ ; here, the frequency spectrum is reduced by 4 elements in each layer, eliminating the frequency components, i.e.,  $s^{(2)} = s^{(4)} = 4$ . Then, the fully connected layer has 35 synaptic weights. For this proposed architecture, the input (57) only take 3 elements for each signal that consist of acceleration at each storey plus acceleration at ground level, velocity, position and hysteretic displacement estimated by the FDCNN.

$$\hat{\omega}(q) = [\hat{y}(q-1), \dots, \hat{y}(q-3), \hat{x}(q-1), \dots, \hat{x}(q-3), \hat{x}(q-1), \dots, \hat{x}(q-3), \hat{x}_g(q), \dots, \hat{x}_g(q-2)]^T \tag{67}$$

Experimental data consist of 11 different tests, from which 9 of them are used for training and two for testing. Each experiment lasts 25 s with sampling time of 5 ms. The excitation signal comes from the Northridge earthquake, shown in Fig. 4, which is adjustment to match with building structure prototype. Experiments consist in using the row data just as it is acquired from the sensors, hoping the FDCNN can deal with the noisy data. Figures 6 and 7 show the results corresponding with the identification of the internal state  $z_i$  of the hysteretic model. From both figures, it is evident that an accurate estimation of the hysteretic state is achieved, since estimate  $z_i$  converges to reference signal. The inclusion

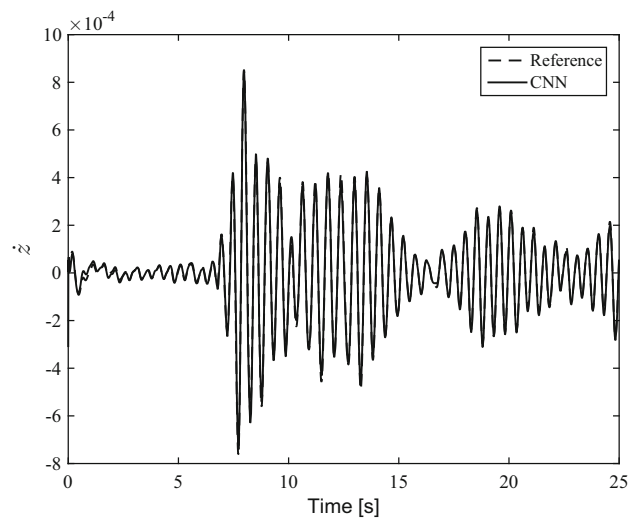


Fig. 6 First-storey hysteretic displacement using FDCNN with noisy data

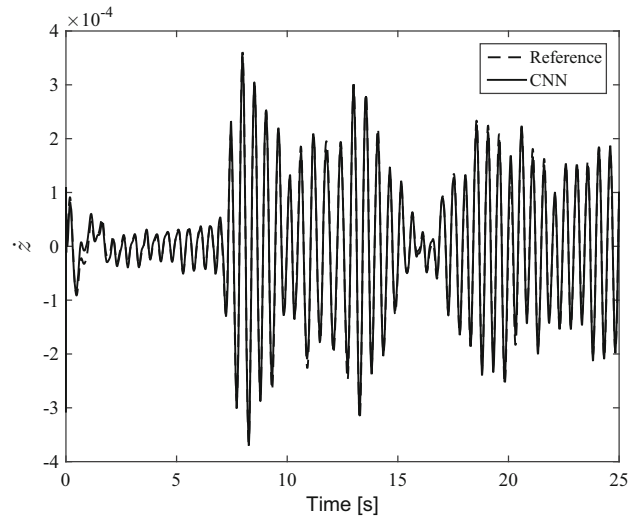


Fig. 7 Second-storey hysteretic displacement using FDCNN with noisy data

of spectral pooling operation in FDCNN eliminates measurement noise and offset in acceleration data. The mean square error obtained is  $2.0901 \times 10^{-9}$ , which is also less than when time domain CNN is used. Computational time is  $3.0164 \times 10^{-9}$  s for a 5-epoch training.

Moreover, note that Figs. 6 and 7 present an oscillatory behavior. Since the seismic excitation signal is oscillatory (harmonic motion), the response measured in each floor is also oscillatory. Therefore, the estimated hysteretic state is also oscillatory because it depends on the estimated velocities at each floor, as defined in Eq. (18).

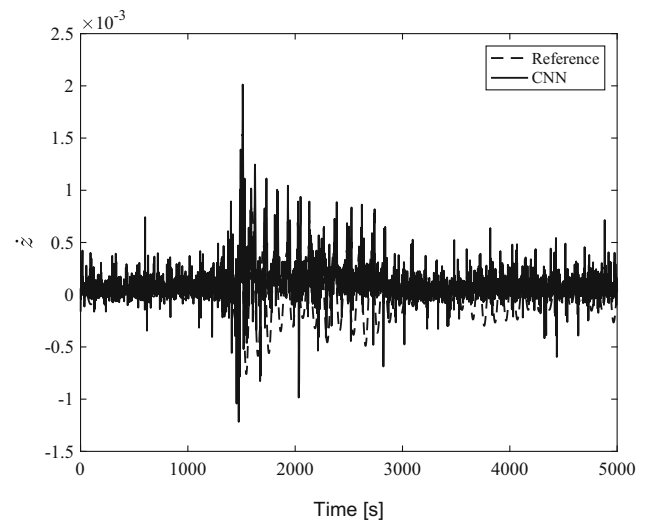
#### 4.1.2 Identification system using time domain CNN

In this subsection, the time domain CNN presented in “Appendix B” is used to identify the hysteretic state  $z_i$  at each storey. For each one of them, a different CNN is used and they do not depend on each other. Further work will focus on the design of a single architecture that describes the complete building dynamics. The seismic excitation used here for data test is also the Northridge earthquake. For the time domain CNN, the hyperparameters are: 2 convolutional layers *Conv1* and *Conv3*, each one with 5 filters,  $h = 5$ , the length of the filters  $f_1 = f_3 = 3$ , and two subsample layers *Sub2* and *Sub4*, in each layer every 2 elements; one is removed and only the one with the highest value is kept,  $s_2 = s_4 = 2$ . Given the proposal architecture, the fully connected layer *Fu5* will have 50 synaptic weights,  $L = 50$ . The learning rate for all the layers is set to 0.3; the input of CNN is a vector building by 4 data of  $\hat{y}$  estimated by the CNN, 8 acceleration data, of which 4 correspond to the excitation in the ground level and the remaining four correspond to the acceleration of each floor. Finally, 4 velocities plus 4 displacement data were also used. Therefore, (77) can be described as (68)

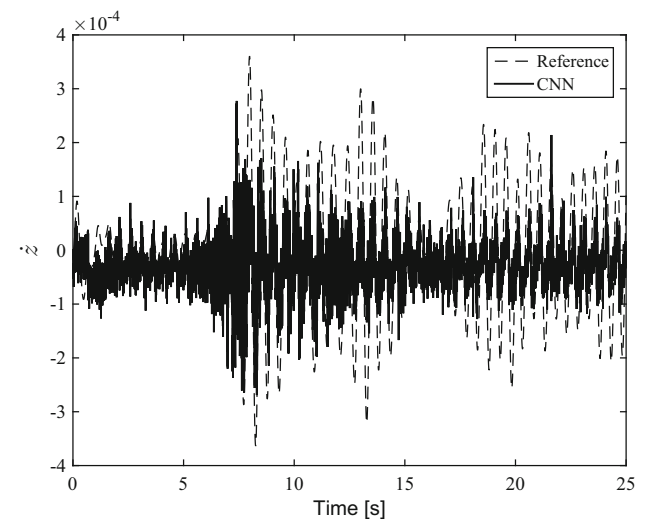
$$\hat{\omega}(q) = [\hat{y}(q-1), \dots, \hat{y}(q-4), \ddot{x}(q-1), \dots, \ddot{x}(q-4), \dot{x}(q-1), \dots, \dot{x}(q-4), \ddot{x}_g(q), \dots, \ddot{x}_g(q-3)]^T \quad (68)$$

Hyperparameters initialization for TDCNN are randomly choose. The output synaptic weights are between  $[-1, 1]$ , and filters in convolutional layers are within the range  $[-\frac{1}{\sqrt{j}}, \frac{1}{\sqrt{j}}]$ , where  $j$  is the length of the input. It is important to point that like in the previous section, experimental data are from 11 tests, of which 9 of them are used for training and two for testing. Each experiment lasts 25 s with sampling time of 5 ms. In order to identify the hysteretic displacement of the building, two different identification tasks were carried out.

(a) The first of them consists in using the row data just as it is acquired from the sensors. Figures 8 and 9 show the identification results of Bouc–Wen hysteretic state for the first and second floors, respectively. From these figures, it can be observed that in both cases the parametric convergence is not achieved, which was expected due to the presence of noise in the measurement and offset. The means square error (MSE) obtained for the first floor is  $1.0127 \times 10^{-7}$  and  $1.5168 \times 10^{-7}$  for the second that is too small because of the magnitude of the signal, which is in order of  $10^{-4}$ . Computational time required in this experiment was 158.52 s for a 5-epoch training.

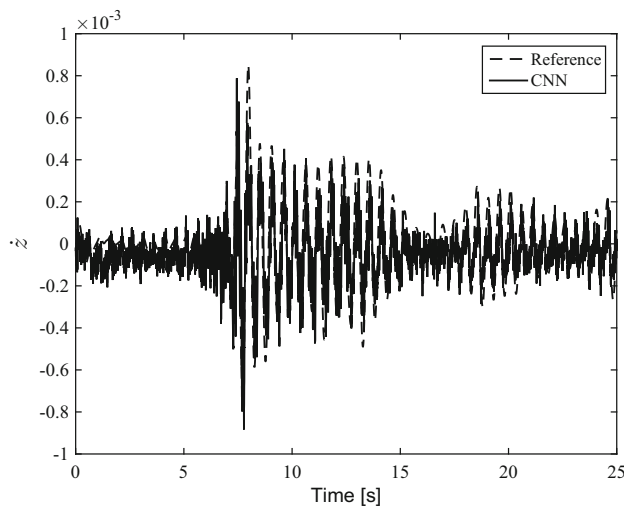


**Fig. 8** First-storey hysteretic displacement using TDCNN with noisy data

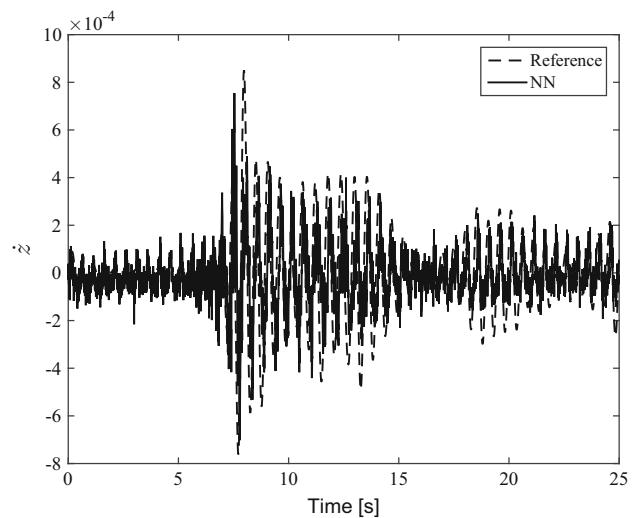


**Fig. 9** Second-storey hysteretic displacement using TDCNN with noisy data

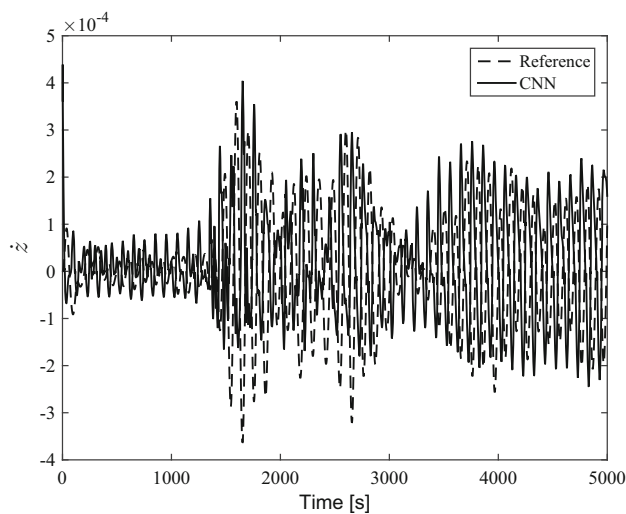
(b) The second identification task consists in using time domain CNN plus filter to eliminate measurement noise and therefore improve the performance of the identification scheme using CNN. The network configuration is the same, previously described in this section. For data processing, a third-order Butterworth filter is added to clean the signal, reducing the components of high and low frequencies. The bandwidth of this filter goes from 0.3 Hz to 5 Hz and was designed in Matlab. However, to get a good performance, a previous knowledge of building bandwidth is required; otherwise, the filter does not reduce the important component of frequency where the system is matching. Experimental results are shown in Figs. 10 and 11. From these figures, it is evident that due to filtering, the estimation of the hysteretic model is improved as shown in Figs. 8 and 9. Despite the



**Fig. 10** First-storey hysteretic displacement using TDCNN with filtered data



**Fig. 12** First-storey hysteretic displacement using NN with noisy data



**Fig. 11** Second-storey hysteretic displacement using TDCNN with filtered data

improvement, convergence is not yet achieved, as a result of the exact lack of bandwidth. However, this situation is so common that it occurs in most systems, because characterization of a building structures is a complicated task. The mean square error (MSE) for stories is  $2.34 \times 10^{-8}$  and  $1.984 \times 10^{-8}$  for the first storey and the second storey, respectively. It is important to note that experiments with 5 epochs of training took 165.7371 s, which is greater than the previous result without filters.

#### 4.1.3 Identification system using neural network (NN)

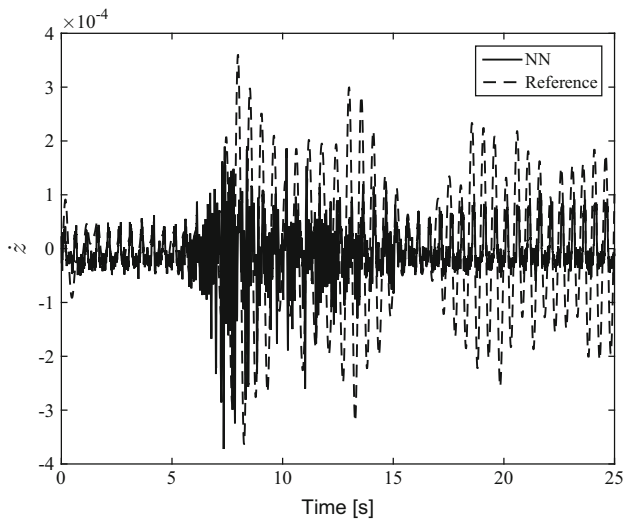
In this subsection, a system identification based on neural network (NN) presented in ‘‘Appendix C’’ is used to estimate the hysteretic displacement  $z_i$ , with  $i = 1, 2$ . Used vibration

data are from 11 tests, of which 9 of them are used for training and two for testing, all of them with duration of 25 s and sampling time of 5 ms. A two-layered neural network (NN) is used for comparisons. Its structure is made up of a hidden layer with 35 nodes, to which a  $\tanh(\cdot)$  activation function is applied; it has only one node in the output layer. The training is done using the BP algorithm with the same amount of data that were used with both CNN methods. In order to identify the hysteretic state at each storey, two different identification tasks were also carried out. The input defined in (65) is also used with the same structure as the one used in the FDCNN.

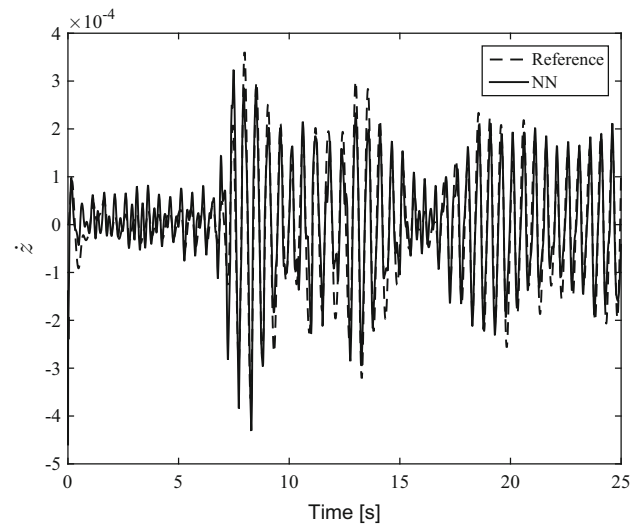
(a) The first of them, like in the previous section, consists in using the row data just as it is acquired from the sensors. Estimated Bouc–Wen hysteretic displacement corresponding to the first floor and second floor is depicted in Figs. 12 and 13, respectively. From these figures, it can be observed that in both cases the estimation does not converge to reference signal, due to measurement noise and offset contained in vibration data.

The means square error (MSE) obtained for the first floor is  $134.05 \times 10^{-10}$  and  $310.54 \times 10^{-10}$  for the second. Computational time required in this experiment was 28.35 s in the worst case, for a 5-epoch training.

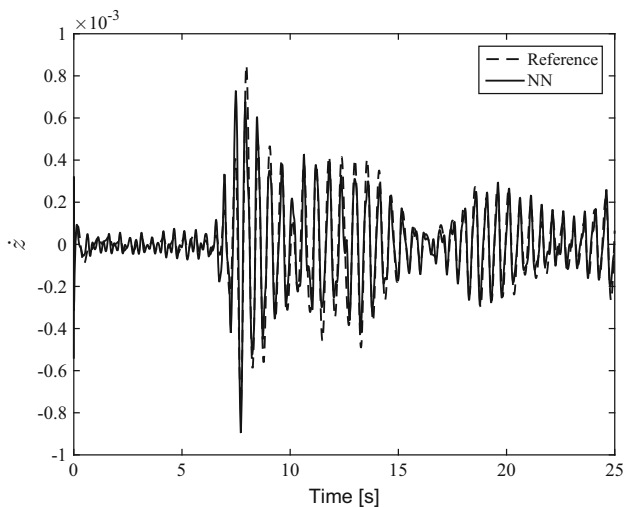
(b) The second identification task consists in using NN plus filter to eliminate measurement noise from vibration data. Data processing was carried out by using a third-order Butterworth filter that reduces the components of high and low frequencies, with bandwidth between 0.3 and 5 Hz. Figures 14 and 15 show that due to filtering, the estimation of the hysteretic model is improved as shown in Figs. 12 and 13. Despite the improvement, convergence is not yet achieved, evidencing that estimated states almost converge to reference signal. The mean square error (MSE) for stories is  $2.34 \times 10^{-8}$  and  $1.984 \times 10^{-8}$  for first storey and sec-



**Fig. 13** Second-storey hysteretic displacement using NN with noisy data



**Fig. 15** Second-storey hysteretic displacement using NN with filtered data



**Fig. 14** First-storey hysteretic displacement using NN with filtered data

ond storey, respectively. This experiment with 5 epochs of training took 27.73 s.

#### 4.1.4 Discussions about identification systems

From obtained results in Sects. 4.1.1, 4.1.2 and 4.1.3, it is evident that the proposed FDCNN has better performance compared to the other two methods; even though the NN is faster in its training, its MSE is higher than the one obtained through FDCNN. In any cases, applying measurement noise in data decreases the performance of the identification methods; however, the FDCNN is barely affected compared to the TDCNN and NN algorithms. Details about these results can be found in Tables 3 and 4, where features like precision, execution time and means square error (MSE) are compared.

Since the structural damage assessment is carried out offline in all cases, the precision is the most important feature for a adequate structural health diagnosis. Thus, the system identification architecture based on FDCNN algorithm is the one has the greatest potential for this task, with the highest precision and considerably low execution time. Therefore, once the versatility of the frequency domain CNN for identification system has been demonstrated under environmental noise, we prefer to use only estimation results from FDCNN for damage detection propose. Hence, in the following section, we will only present damage detection results employing data obtained through FDCNN.

## 4.2 Damage detection in building structure

In this subsection, we investigate damage detection sensibility based on the building capacity to dissipate energy, which is reduced in contrast to nominal conditions. Experiment were carried out reducing the stiffness  $k_2$  on the second storey by loosened only one screw in one of fourth column that make up each level. The remaining 3 columns are not modified. The next step consists in extracting the features of damage building from acceleration measurement, when the prototype was subjected to Northridge earthquake. In consequence, the fundamental vibration frequency and the bandwidth also change due to induced damage, reducing  $f_1 = 1.733$  Hz and  $f_2 = 3.97$  Hz. From vibrational analysis, we know that changes in vibration frequencies are a good indicator for damage detection. Unlike, in this paper we use load-deformation curves and dissipated energy changes for damage assessment and diagnosis. This is achieved, employing the frequency domain CNN for a model-based identification described in Sect. 4.1.1 that allows to confirm structural damage through

**Table 3** Comparison of proposed method with a neural network and TDCNN (first storey)

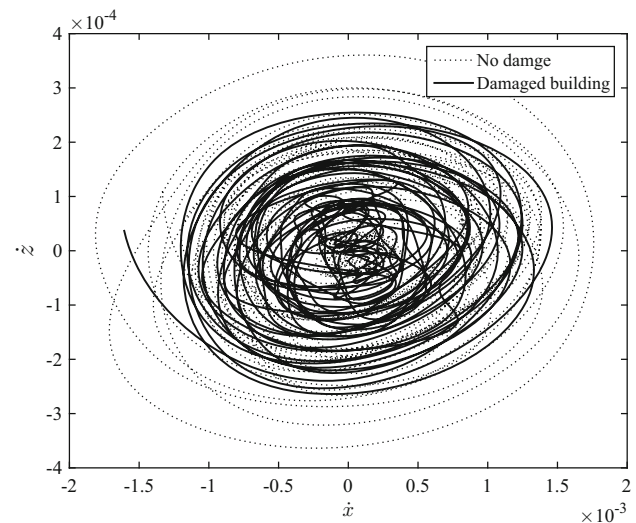
Case	Training time (s)	Testing time (s)	MSE ( $\times 10^{-10}$ )
NN			
No noise	27.73	0.41	22.378
Noise	28.35	0.40	134.05
TDCNN			
No noise	164.41	1.32	313.27
Noise	157.21	1.31	846.71
FDCNN			
No noise	44.24	1.07	1.3182
Noise	42.94	1.04	11.991

**Table 4** Comparison of proposed method with a neural network and TDCNN (second storey)

Case	Training time (s)	Testing time (s)	MSE ( $\times 10^{-10}$ )
NN			
No noise	27.73	0.41	54.484
Noise	28.35	0.40	310.54
TDCNN			
No noise	164.41	1.32	234
Noise	157.21	1.31	1012.7
FDCNN			
No noise	44.24	1.07	2.4728
Noise	42.94	1.04	2.0901

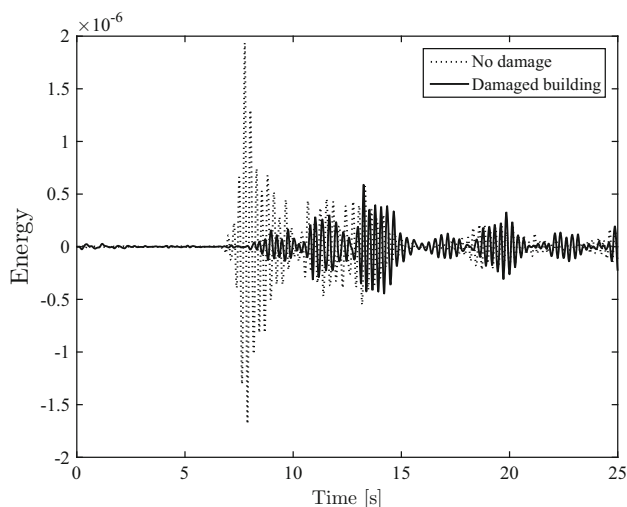
load-deformation curves, obtained after exciting the experimental prototype at basement. A comparison of hysteretic cycles between nominal and damage conditions is shown in Fig. 16 that correspond to the second storey. Results allow to observe that when there is structural damage, the relationship between load and deformation is also reduced significantly, which indicates that the building capacity to dissipate energy is also reduced in contrast to nominal conditions. Moreover, from hysteretic curves we also calculate the dissipated energy to be compared with results in nominal conditions, as shown in Fig. 17 that correspond to the same floor.<sup>1</sup> From Fig. 17, it can be noticed that the energy is much lower in the presence of damage. Results agree with the raised hypothesis of the problem. When there is structural damage, the capacity of the building to dissipate energy is reduced, which indicates that the building changes the elastic to the plastic zone. Results confirm the effectiveness of the proposed identification scheme for damage detection problem, where Bouc–Wen hysteretic model is a useful tool to capture the degrading energy. Similar results are obtained for the first floor.

<sup>1</sup> The energy of the building is estimated using the CNN output together with the velocity of each floor.

**Fig. 16** Second-storey hysteretic cycle

### 4.3 Discussion

Two different perspectives of CNN and one more based on NN were presented applying them to a real problem in two different situations. Tables 3 and 4 compare the obtained results. The training time is one of the most representative results, where the FDCNN performs 4 times faster than the



**Fig. 17** Second-storey energy

TDCNN, and both neural networks have similar architecture, same number of layers and same number of filters, only differ in how the operations are treated in the convolutional and subsample layers. In testing stage, the time does not vary so much because of the size of elements in the operation. Nevertheless, it is possible to appreciate that the proposed frequency domain CNN algorithm improves the execution time, even though the hyperparameters are also initialized in the same interval. In the case of the NN-based scheme, it is faster than FDCNN, but its accuracy in the identification system is less, even when both schemes have similar architecture.

Additionally, in the identification system, FDCNN also has a better performance than TDCNN and NN even when they are accompanied by a filter. Under same conditions and similar neural structure, FDCNN is more suitable for identification system rather than the TDCNN. Perhaps improvements could be present changing the architecture of TDCNN with a deeper structure, which is not possible to achieve with the NN design. Another difference is that the proposed FDCNN does not require any activation function as the TDCNN and NN; this also contributes to the reduction in computational time and does not affect the system identification performance.

## 5 Conclusions

The frequency domain CNN has been proven in this study to be more reliable to use as a identification system rather than time domain CNN in time domain and neural network, as an alternative approach to damage detection in buildings. The results demonstrate that the proposed method is able to learn features from frequency data and achieve higher diagnosis

accuracy. Furthermore, FDCNN introduces the spectral pooling operation in its design that attenuates measurement noise and ensures the convergence of the identification scheme. Note that most methods introduce filters as a previous stage to overcome measurement noise. However, this is difficult to achieve if the system bandwidth is not known in advance, unlike FDCNN that does not need this information. Computational time for FDCNN is almost 4 times faster during the training stage, that is useful for applications with bigger data sets. Moreover, the inclusion of the dissipated energy by using the Bouc–Wen hysteretic model to capture the degrading energy that is directly related to the stiffness loss resulting from structural damage in buildings is an alternative study approach. The use of frequency domain CNN for identification system is an interesting alternative to signal processing method. We also recognized that it is necessary to carry out more and extensive research to assess the potential of this approach. However, we do find the results of the our experimental results to be a good step in that direction.

**Acknowledgements** The authors express their thanks to unknown referees for the careful reading and helpful comments. Authors also appreciate the support of Mr. Jesús Meza for their assistance to complete the experiments. This work was supported in part by the project SEP-CINVESTAV No.62. The second author is also grateful for the financial support of CONACYT. Jesús Morales-Valdez acknowledges the support of Programa Catedras-CONACYT. All authors are grateful to CINVESTAV-IPN for the support in this project.

## Compliance with ethical standards

**Conflict of interest** The authors declare that there is no conflict of interest in this paper.

**Ethical approval** This article does not contain any studies with human participants or animals performed by any of the authors.

## Appendix A: Discrete Fourier transform

Discrete Fourier transform (DFT), denoted by  $\mathcal{F}(\cdot)$ , is a powerful tool to convert spatial samples into a sequence of complex-valued samples in the frequency domain. Some important properties of DFT are as follows: It is linear and unitary (Cooley et al. 1969), and its inverse transform is given by  $\mathcal{F}^{-1}(\cdot) = \mathcal{F}(\cdot)^*$  which is the conjugate of the transform itself. This last property is useful during the training stage of CNN. A DFT of  $n$ -points is defined as  $A = \mathcal{F}(a)$ , where  $\mathcal{F}$  can be expressed as a matrix  $F$ ; this matrix is called a DFT matrix and it is constructed as follows:



$$F_n = \frac{1}{\sqrt{n}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{n-1} \\ 1 & \omega^2 & \omega^3 & \dots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \dots & \omega^{(n-1)(n-1)} \end{bmatrix}$$

where  $\omega = \exp \frac{-2\pi i}{n}$ .

Slight modification is made in order to ensure the DC frequency component in the center row of the matrix.

**Remark 7** In frequency analysis, the convolution operation becomes an element-wise product which makes the analysis easier and direct. The convolution operation between  $a, b \in \mathfrak{R}^n$ , using the DFT is:

$$\mathcal{F}(a * b) = \mathcal{F}(a) \odot \mathcal{F}(b) \tag{69}$$

where  $*$  denotes the convolution operation and  $\odot$  is an element-wise product. This product reduces the number of operations compared to the convolution stage in TDCNN and it make the training process even faster.

### Appendix B: Time domain CNN for modeling time series

Consider an unknown discrete-time nonlinear system

$$y(q) = f(x(q)). \quad x(q + 1) = g(x(q), u(q)) \tag{70}$$

where  $y(q)$  is the scalar output,  $x(q)$  the internal state,  $u(q)$  the input,  $f(\cdot)$  and  $g(\cdot)$  smooth functions,  $f, g \in C^\infty$ .

A nonlinear autoregressive exogenous (NARX) model for (70) is defined as

$$y(q) = \Phi[\varpi(q)] \tag{71}$$

The system dynamics are represented by the unknown nonlinear difference equation  $\Phi$ , where

$$\varpi(q) = [y(q - 1), \dots, y(q - n_y), u(q), \dots, u(q - n_u)]^T \tag{72}$$

$y(q)$  and  $u(q)$  within this equation represent measurable output and input for the system, with  $n_y$  and  $n_u$  the regression order, respectively, which are unknown.

The nonlinear system identification of (71) based on time domain convolutional neural networks (TDCNN) is shown in (73), where  $\hat{y}_T(q)$  is estimation of the real output generated by TDCNN, which is a scalar element.

$$\hat{y}_T(q) = W^{(\ell)T} \vartheta \tag{73}$$

This is fully connected layer with  $W$  as synaptic weights vector and  $\vartheta$  the stacked output of the last subsample layer of TDCNN.

Two more types of layer are introduced in TDCNN. The first layer in TDCNN is a convolutional one, where two operations are made: convolution and an activation function. The convolution operation is

$$\chi_h^{(\ell)} = K_h * y_h^{(\ell-1)} \tag{74}$$

$\ell$  represent the actual layer,  $h$ -filters per layer are used, and each filter is  $K_h^{(\ell)} \in R^{f_\ell}$ . For each element  $i$  of  $\chi_h^{(\ell)}$ , the previous operation is equivalent to

$$\chi_{i,h}^{(\ell)} = \sum_{a=0}^{f_\ell-1} K_{h,a}^{(\ell)} y_{h,i+a}^{(\ell-1)} \tag{75}$$

The result of this operation  $\chi_h^{(\ell)}$  is called the feature map, which contains features properties of the input, and each filter obtains a different feature. These feature maps go through an activation function, different activation functions are used in neural networks for specific tasks and unique properties (Glorot and Bengio 2010), but the one used is in this papers is the rectified linear unit (ReLU). The output of a convolutional layer is defined by (76)

$$y_h^{(\ell)} = \max(0, \chi_h^{(\ell)}) \tag{76}$$

for the first layer of the CNN,  $y_h^{(\ell-1)}$  is the input vector

$$\hat{\varpi}(q) = [\hat{y}(q - 1), \dots, \hat{y}(q - r_1), u(q), \dots, u(q - r_2)]^T \tag{77}$$

where  $r_1$  and  $r_2$  denote the regression order.  $r_1 \neq n_y$  and  $r_2 \neq n_u$ .

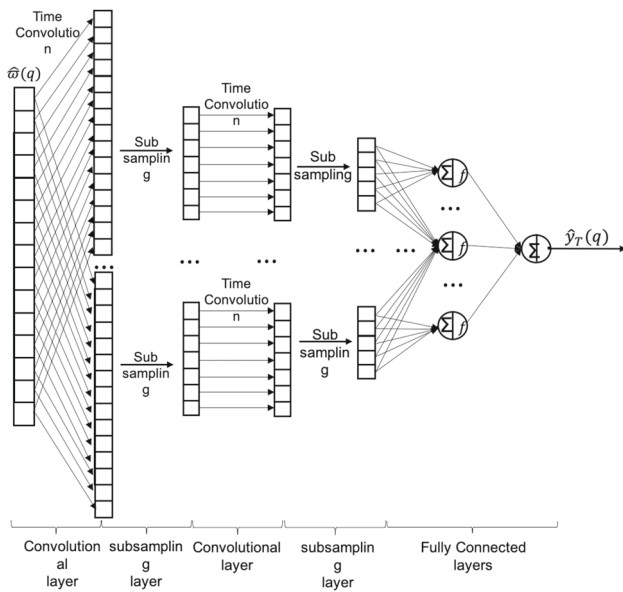
After a convolutional layer, a subsample layer is followed; this layer is pretended to be used as data reduction stage, so the strongest response from the filters keeps going through the TDCNN.

In the subsample layers, the operation used is the max-pool, which is defined as

$$y_h^{(\ell)} = \maxpool(y_h^{(\ell-1)}, s_\ell) \tag{78}$$

The input divided in groups of dimension  $s_\ell$  and from each group the highest values remain. The Shrink depends on the layer where it is applied.

Convolutional and subsample layers can be repeated as many times as the application require in the TDCNN. As mentioned earlier, after the last subsample layer, the outputs of each feature map are stacked to create the vector  $\vartheta$



**Fig. 18** Time domain convolutional neural network for system identification

$$\vartheta = [y_1^{(\ell)T} \ y_2^{(\ell)T} \ \dots \ ; \ y_h^{(\ell)T}]^T \tag{79}$$

This helps to manage the last layer in terms of vector and matrices. The complete architecture is shown in Fig. 18.

**Training of time domain CNN using backpropagation**

The training of the TDCNN’s parameters is realized by the backpropagation algorithm (BPA), which is used to calculate the gradient of the cost function respect each parameter of the TDCNN, propagating it backward through the network to update these parameters. The cost function is used as a measurement of the performance, and the most frequently cost function for identification is the squared error which measures the difference between the real output and the estimated one.

$$J(q) = \frac{1}{2} e_T^2(q) \tag{80}$$

$e_T(q)$  is the identification error between the TDCNN output and the real output in each instant, i.e.,  $e_T(q) = \hat{y}_T(q) - y(q)$ .

The BPA uses the gradient of the cost function with respect to each parameter in the neural network. To calculate the gradient, it uses the chain rule and then each parameter is updated by the delta rule. In the output layer, the weights are updated as follows:

$$w_i^{(\ell)}(q + 1) = w_i^{(\ell)}(q) - \eta_T \frac{\partial J}{\partial w_i^{(\ell)}} \tag{81}$$

where  $w_i^{(\ell)}$  are the elements of the vector  $W^{(\ell)}$ ,  $\eta_T$  the learning rate defining one for each layer and

$$\frac{\partial J}{\partial w_i^{(\ell)}} = \frac{\partial J}{\partial e_T} \frac{\partial e_T}{\partial \hat{y}_T} \frac{\partial \hat{y}_T}{\partial w_i^{(\ell)}} = e_T \vartheta_i \tag{82}$$

$\vartheta_i$  are the elements of vector  $\vartheta$  corresponding to the weight  $w_i^{(\ell)}$ . To previous layer, the gradient, using chain rule, is

$$\frac{\partial J}{\partial \vartheta} = \frac{\partial J}{\partial e_T} \frac{\partial e_T}{\partial \hat{y}_T} \frac{\partial \hat{y}_T}{\partial \vartheta} = e_T W^{(\ell)} \tag{83}$$

For the subsample layer, an reverse operation of *maxpool* is used to calculate the gradient

$$\frac{\partial J}{\partial y^{(\ell-1)}} = up \left( \frac{\partial J}{\partial y^{(\ell)}} \right) \tag{84}$$

where  $up(\cdot)$  is an operation to increase length of the gradient to match the previous layer and only passing to the positions where the highest response occurs in the forward stage, leaving everything else in zeros. For convolutional layer, the gradient of the cost function with respect to the filters is calculated as

$$\frac{\partial J}{\partial K_h^{(\ell)}} = y_h^{(\ell-1)} * rot180(\delta_h^{(\ell)}) \tag{85}$$

with  $*$  being the convolution operator and

$$\delta_{h,i}^{(\ell)} = \frac{\partial J}{\partial y_{h,i}^{(\ell)}} f'(\chi_{h,i}^{(\ell)}) \tag{86}$$

with  $f'(\cdot)$  being the derivative of the *ReLU* operation, that is defined as,

$$f'(\Omega) = \begin{cases} 1 & \text{if } \Omega > 0 \\ 0 & \text{otherwise} \end{cases}$$

In order to update the filters, delta rule is used, therefore

$$K_h^{(\ell)}(q + 1) = K_h^{(\ell)}(q) - \eta_T \left( y_h^{(\ell-1)} * rot180(\delta_h^{(\ell)}) \right) \tag{87}$$

Finally, to backpropagate the gradient to previous layer of a convolutional layer, the equation is

$$\frac{\partial J}{\partial y_h^{(\ell-1)}} = \delta_h^{(\ell)} \odot rot180(K_h^{(\ell)}) \tag{88}$$

The operator  $rot180(\cdot)$  is equivalent to use its parameter from bottom to top, just like a flip over.

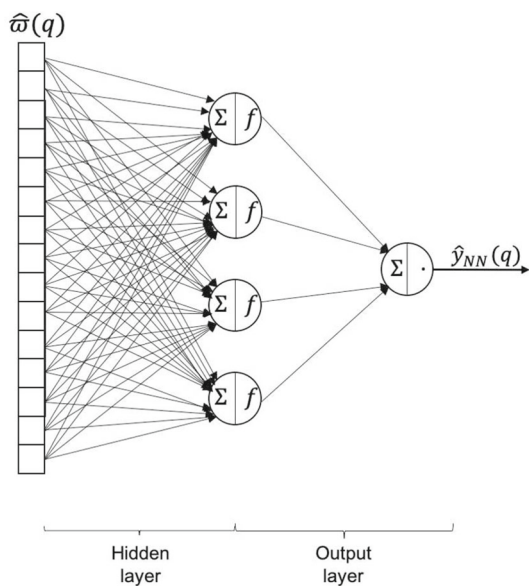


Fig. 19 Multilayer perceptron architecture for system identification

### Appendix C: Multilayer neural network for system modeling

For comparison, a multilayer perceptron (NN for simplicity) is created. This NN consists of one hidden layer with 35 units, which are paired with activation function  $\tanh(\cdot)$ ; its architecture is shown in Fig. 19.

Consider the nonlinear system to be identify defined in 71 and regard the same input from the CNN described in 35; the output of the units in the hidden layer is defined as:

$$X_{NN} = V_{NN}\varpi \tag{89}$$

$V_{NN}$  are the synaptic weights in the hidden layers written in matrix form,  $X_{NN}$  is the vector output of hidden layer, and each element corresponds to each one of the units in this layer. The output of the NN is

$$\hat{y}_{NN} = W_{NN}X_{NN} \tag{90}$$

where  $W_{NN}$  are the synaptic weights in the output layer, dimensions match, so the output is scalar. The training of this NN is done with the backpropagation algorithm. For this matter, the cost function to be minimized is defined as

$$J(q) = \frac{1}{2}e_{NN}(q)^2 \tag{91}$$

where  $e(q) = (\hat{y}_{NN}(q) - y(q))^2$  and the update law for the synaptic weights in output and hidden layer is defined with the delta rule, i.e.,

$$W_{NN}(q + 1) = W_{NN}(q) - \eta_{NN} \frac{\partial J}{\partial W_{NN}} \tag{92}$$

and

$$V_{NN}(q + 1) = V_{NN}(q) - \eta_{NN} \frac{\partial J}{\partial V_{NN}} \tag{93}$$

where  $\eta_{NN}$  is the learning rate for this NN.

### References

Abdeljaber O, Avci O, Kiranyaz S, Gabbouj M, Inman DJ (2017) Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *J Sound Vib* 388:154–170

Appana DK, Prosvirin A, Kim J-M (2018) Reliable fault diagnosis of bearings with varying rotational speeds using envelope spectrum and convolution neural networks. *Soft Comput* 22(20):6719–6729

Arqub OA, Abo-Hammour Z (2014) Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm. *Inf Sci* 279:396–415

Arqub OA, Mohammed AL-S, Momani S, Hayat T (2016) Numerical solutions of fuzzy differential equations using reproducing kernel Hilbert space method. *Soft Comput* 20(8):3283–3302

Arqub OA, Al-Smadi M, Momani S, Hayat T (2017) Application of reproducing kernel algorithm for solving second-order, two-point fuzzy boundary value problems. *Soft Comput* 21(23):7191–7206

Atha DJ, Jahanshahi MR (2018) Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection. *Struct Health Monit* 17(5):1110–1128

Bouti A, Mahraz MA, Riffi J, Tairi H (2018) A robust system for road sign detection and classification using lenet architecture based on convolutional neural network. *Soft Comput* 24:6721–6733

Bursi OS, Ceravolo R, Erlicher S, Zanotti Fragonara L (2013) Identification of the hysteretic behaviour of a partial-strength steel-concrete moment-resisting frame structure subject to pseudodynamic tests. *Earthq Eng Struct Dyn* 41(14):1883–1903

Carden EP, Fanning P (2004) Vibration based condition monitoring: a review. *Struct Health Monit* 3:355–377

Ceravolo R, Erlicher S, Fragonara LZ (2013) Comparison of restoring force models for the identification of structures with hysteresis and degradation. *J Sound Vib* 332(26):6982–6999

Cha Y-J, Choi W, Büyüköztürk O (2017) Deep learning-based crack damage detection using convolutional neural networks. *Comput Aided Civ Infrastruct Eng* 32(5):361–378

Charles RF, Keith W, Michael DT, Gyuhae P, Jonathon N, Douglas EA, Matthew TB, Kevin F (2007) Nonlinear system identification for damage detection. In: Report LA-14353, Los Alamos National Laboratory (LANL), Los Alamos, NM, pp 1–161

Chatzi EN, Smyth AW, Masri SF (2010) Experimental application of on-line parametric identification for nonlinear hysteretic systems with model uncertainty. *Struct Saf* 32(5):326–337

Chen S, Billings S, Grant P (1990) Non-linear system identification using neural networks. *Int J Control* 51(6):1191–1214

Chopra AK (1995) Dynamics of structures: theory and applications to earthquake engineering, 1st edn. Prentice-Hall International series

Cooley JW, Lewis PA, Welch PD (1969) The fast Fourier transform and its applications. *IEEE Trans Educ* 12(1):27–34

Das S, Saha P, Patro S (2016) Vibration-based damage detection techniques used for health monitoring of structures: a review. *J Civ Struct Health Monit* 6(3):477–507

- Doebling SW, Farrar C, Prime MB (1998) A summary review of vibration-based damage identification methods. *Shock Vib Dig* 30(2):1–34
- Eslami E, Choi Y, Lops Y, Sayeed A (2019) A real-time hourly ozone prediction system using deep convolutional neural network. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-019-04282-x>
- Fan W, Qiao P (2011) Vibration-based damage identification methods: a review and comparative study. *Struct Health Monit* 10:83–111
- Farrar C, Doebling S, Nix D (2001) Vibration-based structural damage identification. *Philos Trans R Soc* 359(1778):131–149
- Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp 249–256
- Hibberler RC (2011) *Mechanics of materials*, 8th edn. Prentice Hall, pp 1–888
- Ikhrouane FA, Mañosa VC, Rodellar J (2005) Adaptive control of a hysteretic structural system. *Automatica* 41(2):225–231
- Kim Y (2014) Convolutional neural networks for sentence classification. [arXiv:1408.5882](https://arxiv.org/abs/1408.5882)
- Kong X, Cai C-S, Hu J (2017) The state-of-the-art on framework of vibration-based structural damage identification for decision making. *Appl Sci* 7(5):497–510
- Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*, pp 1097–1105
- Lawrence S, Giles CL, Tsoi AC, Back AD (1997) Face recognition: a convolutional neural-network approach. *IEEE Trans Neural Netw* 8(1):98–113
- LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. *Neural Comput* 1(4):541–551
- Lin Y-Z, Nie Z-H, Ma H-W (2017) Structural damage detection with automatic feature-extraction through deep learning. *Comput Aided Civ Infrastruct Eng* 32(12):1025–1046
- Liu R, Yang B, Zio E, Chen X (2018a) Artificial intelligence for fault diagnosis of rotating machinery: a review. *Mech Syst Signal Process* 108:32–47
- Liu Y, Huang H, Cao J, Huang T (2018b) Convolutional neural networks-based intelligent recognition of chinese license plates. *Soft Comput* 22(7):2403–2419
- Loh C-H, Mao C-H, Huang J-R, Pan T-C (2011) System identification and damage evaluation of degrading hysteresis of reinforced concrete frames. *Earthq Eng Struct Dyn* 40(6):623–640
- Ma F, Zhang H, Bockstedte A, Foliente GC, Paevere P (2004b) Parameter analysis of the differential model of hysteresis. *J Appl Mech* 71(3):342–349
- Ma F, Ng CH, Ajavakom N (2006) On system identification and response prediction of degrading structures. *Struct Control Health Monit* 13:347–364
- Ma S, Cai W, Liu W, Shang Z, Liu G (2019) A lighted deep convolutional neural network based fault diagnosis of rotating machinery. *Sensor* 19(10):2381
- Maia NMM, Silva JMM, Almas EAM, Sampaio RPC (2003) Damage detection in structures: from mode shape to frequency response function methods. *Mech Syst Signal Process* 17(3):489–498
- Modarres C, Astorga N, Drogue EL, Meruane V (2018) Convolutional neural networks for automated damage recognition and damage type identification. *Struct Control Health Monit* 25:e2230
- Pau A, Vestroni F (2013) Vibration assessment and structural monitoring of the basilica of maxentius in rome. *Mech Syst Signal Process* 41:454–466
- Rahai A, Bakhtiari-Nejad F, Esfandiari A (2007) Damage assessment of structure using incomplete measured mode shapes. *Struct Control Health Monit* 14:808–829
- Rippel O, Snoek J, Adams RP (2015) Spectral representations for convolutional neural networks. In: *Advances in neural information processing systems*, pp 2449–2457
- Roux P, Guéguen P, Baillet L, Hamze A (2014) Structural-change localization and monitoring through a perturbation-based inverse problem. *Acoust Soc Am* 136:2586–2597
- Rucevskis S, Janeliukstis R, Akishin P, Chate A (2016) Mode shape-based damage detection in plate structure without baseline data. *Struct Control Health Monit* 23:1180–1193
- Shan J, Shi W, Lu X (2016a) Model-reference health monitoring of hysteretic building structure using acceleration measurement with test validation. *Comput Aided Civ Infrastruct Eng* 31:449–464
- Shin M, Lee J-H (2016) Cnn based lithography hotspot detection. *Int J Fuzzy Log Intell Syst* 16(3):208–215
- Simard PY, Steinkraus D, Platt JC, et al (2003) Best practices for convolutional neural networks applied to visual document analysis. In: *ICDAR*, vol 3
- Sohn H, Farrar C, Hemez F, Devin DS, Daniel WS, Brett RN, Jerry JC (2003) A review of structural health monitoring literature: 1996–2001. In: *Los Alamos National Laboratory report*, LA-13976-MS, pp 1–331
- Udmale SS, Patil SS, Phalle VM, Singh SK (2019) A bearing vibration data analysis based on spectral kurtosis and convnet. *Soft Comput* 23(19):9341–9359
- Vidal F, Navarro M, Aranda C, Enomoto T (2014) Changes in dynamic characteristics of lorca rc buildings from pre- and post-earthquake ambient vibration data. *Bull Earthq Eng* 12:2095–2110
- Vu T-D, Ho N-H, Yang H-J, Kim J, Song H-C (2018) Non-white matter tissue extraction and deep convolutional neural network for alzheimer's disease detection. *Soft Comput* 22(20):6825–6833
- Wan Z, Wang T, Li S, Zhang Z (2018) A modified particle filter for parameter identification with unknown inputs. *Struct Control Health Monit* 25:e2268
- Wen YK (1976) Method for random vibration of hysteretic system. *J Eng Mech Div* 102(2):249–263
- Yıldırım Ö, Baloglu UB, Acharya UR (2018) A deep convolutional neural network model for automated identification of abnormal eeg signals. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-018-3889-z>
- Zhao R, Yan R, Chen Z, Mao K, Wang P, Gao RX (2019) Deep learning and its applications to machine health monitoring. *Mech Syst Signal Process* 115:213–237
- Zhu H, Li L, He X-Q (2011) Damage detection method for shear buildings using the changes in the first mode shape slopes. *Comput Struct* 89(9–10):733–743
- Zou Y, Tong L, Steven GP (2000) Vibration-based model dependent damage (delamination) identification and health monitoring for composite structures: a review. *J Sound Vib* 230:357–378