



MOX-Report No. 63/2021

**Online structural health monitoring by model order  
reduction and deep learning algorithms**

Rosafalco, L.; Torzoni, M.; Manzoni, A.; Mariani, S.;  
Corigliano, A.

MOX, Dipartimento di Matematica  
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

[mox-dmat@polimi.it](mailto:mox-dmat@polimi.it)

<http://mox.polimi.it>

# Online Structural Health Monitoring by Model Order Reduction and Deep Learning Algorithms

Luca Rosafalco<sup>1</sup>, Matteo Torzoni<sup>1</sup>, Andrea Manzoni<sup>2</sup>, Stefano Mariani<sup>1</sup>, and Alberto Corigliano<sup>1</sup>

<sup>1</sup>Dipartimento di Ingegneria Civile e Ambientale, Politecnico di Milano, Italy\*

<sup>2</sup>MOX, Dipartimento di Matematica, Politecnico di Milano, Italy<sup>†</sup>

## Abstract

Within a structural health monitoring (SHM) framework, we propose a simulation-based classification strategy to move towards online damage localization. The procedure combines parametric Model Order Reduction (MOR) techniques and Fully Convolutional Networks (FCNs) to analyze raw vibration measurements recorded on the monitored structure. First, a dataset of possible structural responses under varying operational conditions is built through a physics-based model, allowing for a finite set of predefined damage scenarios. Then, the dataset is used for the offline training of the FCN. Because of the extremely large number of model evaluations required by the dataset construction, MOR techniques are employed to reduce the computational burden. The trained classifier is shown to be able to map unseen vibrational recordings, *e.g.* collected on-the-fly from sensors placed on the structure, to the actual damage state, thus providing information concerning the presence and also the location of damage. The proposed strategy has been validated by means of two case studies, concerning a 2D portal frame and a 3D portal frame railway bridge; MOR techniques have allowed us to respectively speed up the analyses about 30 and 420 times. For both the case studies, after training the classifier has attained an accuracy greater than 85%.

*Keywords:* structural health monitoring, deep learning, reduced order models, fully convolutional networks, damage localization.

## 1 Introduction

Physics-based models are derived from first principles expressing the laws of nature, and from accurate and efficient numerical methods for their approximation, such as Finite Elements (FE). Dual to the physics-based paradigm is the data-driven paradigm, according to which laws and patterns governing complex data systems are unveiled by using statistical tools, among which Machine Learning (ML) stands out. Physics-based and data-driven paradigms are not mutually exclusive, even if in the past they were mostly applied separately. Indeed, when dealing with data assimilation, data-driven modelling might be exploited to establish a link between physics-based models and experimental data [1, 2]. Structural Health Monitoring (SHM) is one of the research fields in which this combination is more promising [3].

---

\*{luca.rosafalco,matteo.torzoni,stefano.mariani,alberto.corigliano}@polimi.it

<sup>†</sup>andrea1.manzoni@polimi.it

SHM aims at detecting, locating and quantifying the inception and propagation of damage in a structure by analysing data acquired through pervasive sensor networks [4]. As said, the above paradigms yield two frameworks in SHM: the *model-based* and the *data-driven* approaches. Model-based approaches deal with the monitoring of civil infrastructures by relying on the update of a numerical model (*e.g.* through Kalman filters [5, 6, 7] or optimization procedures [8]). They enable a mechanical intuition of the structural degradation process and the forecasting of the system evolution (prognosis). On the other hand, model-based approaches hardly manage to deal with the great amount of (noisy) data acquired through sensor networks. For this reason, data-driven approaches have become more and more widespread [9, 10].

Among data-driven approaches, we can distinguish supervised [11] and unsupervised [12, 13, 14, 15, 16] methods. Supervised methods employ labeled data referring both to the undamaged condition, assumed as baseline, and to the damage scenarios possibly affecting the structure. Unsupervised methods rely only on unlabeled data collected from the undamaged condition. When coming to damage localization and quantification, supervised methods are more powerful, although they feature an obvious drawback: experimental data referring to possible damage conditions of the structure are, a priori, not available.

To cope with this issue, model-based and data-driven approaches are combined by introducing a physics-based model to simulate the effect of damage on the dynamic response of the structure. Specifically, SHM is approached as a classification problem [17] and different damage scenarios, featuring a range of damage classes, are numerically simulated and used as dataset to train a ML-based classifier. This approach is called Simulation-Based Classification (SBC) [18, 19, 20]. To be effective, the construction of the dataset should take into account the effect of varying operational and environmental conditions. In this respect, Model Order Reduction (MOR) techniques for parametrized systems, such as the Reduced Basis (RB) method [21], can be exploited to speed up the generation of the dataset, thus replacing the solution of a high-fidelity, Full Order Model (FOM) with a cheaper, yet accurate, approximation obtained through a Reduced Order Model (ROM). The RB method provides a low-dimensional approximation of the set of solutions of the FOM, within a prescribed parameters range. It combines a handful of FOM solutions (or snapshots) computed for a set of parameter values to generate a low dimensional space exploiting, *e.g.*, the Proper Orthogonal Decomposition (POD). After the ROM has been built (offline), the associated approximation for the input parameters within the range of interest can be obtained (online) in an almost inexpensive way.

In this paper, a ML-based classifier has been designed by using concepts typical of Deep Learning (DL), a branch of ML: the processing of raw data and the classification task are enveloped through the minimization of a single loss function [22] by exploiting a suitable Neural Network (NN) architecture [23, 24], called Fully Convolutional Network (FCN). This latter is trained to assign labels (or classes) to input data given by vibrational measurements of the monitored structure. The strength of this architecture, already successfully applied in [25, 20], relies on the repeated convolution operations. It has been proven to be much less resource-demanding than the feed-forward NN one; as it is also tailored to detect the correlation within a time series and also across different time series, it looks appropriate for SHM purposes as correlations are actually induced by the vibrations of the structure excited by the external loads. Alternative architectures have been recently proposed to approximate the solution of partial differential equations, by adopting the energy of the system as a "*natural loss function for a machine learning method*", see [26].

While in the past MOR techniques [18, 19] and DL [24] have been explored separately, here their use is advantageously combined to exploit the physical knowledge of the system, to minimize the efforts and time required by data processing, but also to efficiently cope with uncertainties (due to, *e.g.*, the operational conditions of the structure). Moreover, by constructing the training dataset and by training the classifier offline, we allow for an

extremely efficient online monitoring of the structure, given that the trained NN only needs to apply a fixed sequence of linear mappings and nonlinear (activation) functions to the incoming vibration recordings to obtain the classification outcome.

The assessment of the proposed procedure has been done through two numerical case studies addressing several operative issues, from the influence of the ROM reconstruction error on the classification accuracy, to the effect of different measurement noise levels on the procedure performance. Experimental data, still not investigated in this paper, will be addressed in future works.

The remainder of the paper is organized as follows. In Sec. 2, we detail our methodology, highlighting the stochastic treatment of the operational and damage conditions faced by the structure, the construction of the (reduced order) physics-based model, and the setting of the FCN. In Secs. 3 and 4 two numerical case studies are discussed, dealing with a two-dimensional portal frame and an integral concrete portal frame railway bridge, respectively. Conclusions, remarks and future developments are finally discussed in Sec. 5.

## 2 Methodology

In the following, we present our methodology. Specifically: in Sec. 2.1 we detail the content of the dataset  $\mathbf{D}$  used to train and validate the classifier  $\mathcal{G}$ ; in Sec. 2.2 we focus on the high-fidelity FOM of the structure; in Sec. 2.3 we show how a ROM is next obtained through a MOR technique; finally, in Sec. 2.4, we discuss the peculiarities of the employed NN architecture.

### 2.1 Dataset definition

A set of  $N_0$  sensors is exploited to track the vibrational response of the monitored structure [27, 28, 29]. Within the observation interval  $(0, T)$ , each sensor is assumed to provide  $L$  measurements of the local displacement with fixed sampling rate, all collected in the vector  $\mathbf{u}_n \in \mathbb{R}^L$  ( $n = 1, \dots, N_0$ ); the same holds if accelerations  $\ddot{\mathbf{u}}_n$  are sensed. Here, we consider only displacement measurements, even if (see Sec. 3-4) our methodology can cope with acceleration recordings too. Obviously, both the duration  $T$  of the time interval and the sampling rate must be chosen according to the structural frequencies to be handled and to the sensor characteristics.

We call *instance* a set of recordings  $\mathbf{U}_i = [\mathbf{u}_1^i, \dots, \mathbf{u}_{N_0}^i] \in \mathbb{R}^{L \times N_0}$ ,  $i = 1, \dots, I$ , related to the same time interval. The dataset  $\mathbf{D}$ , used to train and validate the classifier  $\mathcal{G}$ , is constructed by collecting  $I$  instances

$$\mathbf{D} = \{(\mathbf{U}_1, g_1), \dots, (\mathbf{U}_I, g_I)\}, \quad (1)$$

where:  $I = I_{tr} + I_{val}$ ;  $g_i$  labels the damage state (if any) in the structure.

The goal of the classifier  $\mathcal{G}$  is to build the underlying mapping between  $\mathbf{U}_i$  and  $g_i$  [30]. During the training phase (see Sec. 2.4),  $I_{tr}$  instances are employed by the classifier to *learn* how to model this mapping [31], while  $I_{val}$  instances are later used to validate the learning process by verifying that the training data are not simply *memorized*. Once trained, the classifier should be able to map an unseen instance  $\mathbf{U}_i$ , uniquely defined by the damage state  $g_i = 0, \dots, G$  ( $g_i = 0$  refers to the undamaged condition) and by the operational conditions, into the correct damage class  $g_i$ . Only a finite number  $G$  of damage states is allowed for, coherently with the classification framework within which we approach the SHM problem. Both the possible damage states and operational conditions must be determined through a preliminary study, by evaluating the mechanical behavior of the structure.

The vector of parameters  $\boldsymbol{\eta}_i \in \mathbb{H} \subset \mathbb{R}^Q$  (*e.g.* acting as load multipliers) is used to describe the operational conditions relevant to the  $i$ -th instance. We assume that  $\boldsymbol{\eta}_i$  does not vary in  $(0, T)$ ; in other words, a time independent set of parameters is associated to

each instance. Even the damage state  $g_i$  is assumed to be frozen within the time interval of interest, coherently with the damage growth typically faced by a civil structure [5]. The  $q$ -th parameter  $\eta_q^i$  ( $q = 1, \dots, Q$ ) is sampled from a continuous probability density function (pdf)  $\mathcal{P}_q$ , preliminarily set (see, *e.g.*, [32, 33], where the authors talk about *probabilistic input data*). Similarly, the occurrence of the considered damage state is sampled from a discrete pdf  $\mathcal{P}_g$ . Different sampling strategies, both random (*e.g.* Latin Hypercube [34]) and quasi-random (*e.g.* Sobol' sequences [35]) can be adopted to explore the parametric space defined by the combination between the parameters governing the operational conditions and the damage states. In this work, a Latin Hypercube Sampling (LHS) has been adopted, as it provided a good compromise between randomness and coverage of the parameter domain [36]. The instance  $\mathbf{U}_i$ , corresponding to the sampled  $\{g, \boldsymbol{\eta}\}_i$ , is simulated through a numerical model.

To slightly simplify the notation, in the following the index  $i$  relevant to the instance will be dropped. Anyhow, it must be remembered that the discussion refers to the  $i$ -th instance only, and computations must be therefore repeated  $I$  times at varying loading and operational conditions.

## 2.2 Full Order Model construction

Before coming to the details related to the FOM setting, few hypotheses concerning the response of a civil infrastructure under varying operational conditions are discussed. The strains and displacements are assumed to be small and, if not specified otherwise, damping effects are disregarded; see *e.g.* [5, 37] for some results regarding the relevance of damping in the identification of continuously excited structures. Damage is modeled as a localized reduction in stiffness, temporarily frozen in time. Despite the simplicity of this last assumption, which rests on a time scale separation between damage evolution and health assessment, many engineering problems can be tackled as discussed, *e.g.*, in [38].

To describe the behavior of the structure, we rely upon linear elasto-dynamics. By space discretizing the governing equation through Finite Elements (FEs), we obtain the following semi-discretized problem:

$$\begin{cases} \mathbf{M}\ddot{\mathbf{v}} + \mathbf{K}(g)\mathbf{v} = \mathbf{f}(\boldsymbol{\eta}), & t \in (0, T) \\ \mathbf{v}(0) = \mathbf{v}_0(\boldsymbol{\eta}) \\ \dot{\mathbf{v}}(0) = \dot{\mathbf{v}}_0(\boldsymbol{\eta}) \end{cases} \quad (2)$$

where:  $\mathbf{v} = \mathbf{v}(t) \in \mathbb{R}^M$  is the displacement vector, while  $\dot{\mathbf{v}}$  and  $\ddot{\mathbf{v}}$  are the corresponding velocity and acceleration vectors;  $\mathbf{M} \in \mathbb{R}^{M \times M}$  is the mass matrix;  $\mathbf{K}(g) \in \mathbb{R}^{M \times M}$  is the elastic stiffness matrix;  $\mathbf{f}(\boldsymbol{\eta})$  is the vector collecting the external loadings. Here  $M$  denotes the total number of degrees of freedom (dofs) of the FE space.

A time discretization of the monitoring window  $(0, T)$  is next determined on the basis of the sensor sampling rate. By adopting a suitable time integration scheme (like the generalised- $\alpha$  method [39]), the displacements  $\mathbf{v}_l$  (with  $l = 0, \dots, L$ ) related to the sampled  $\{g, \boldsymbol{\eta}\}$  are provided. Displacements  $\mathbf{v}_l$  are then all collected in  $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_L] \in \mathbb{R}^{M \times L}$  and, through a Boolean matrix  $\mathbf{T} \in \mathbb{R}^{N_0 \times M}$ , whose  $(n, m)$ -th entry is equal to 1 if and only if the position and orientation in space of the  $n$ -th sensor and of the  $m$ -th dof coincide, the corresponding instance

$$\mathbf{U} = (\mathbf{T}\mathbf{V})^T, \quad (3)$$

is obtained.

## 2.3 Reduced Order Model construction

To get a high quality dataset  $\mathbf{D}$ , the number of required instances may be extremely high. By increasing  $I$ , we enhance the performance of  $\mathcal{G}$  even if, beyond a certain threshold,

the gain becomes marginal. To speed up the dataset construction, we propose to adopt parametric MOR techniques [21, 40] as detailed below.

The FOM solution is approximated as  $\mathbf{v} \approx \mathbf{W}\mathbf{v}_R$ ,  $\mathbf{v}_R \in \mathbb{R}^W$ , through a linear combination of  $W \ll M$  basis functions  $\mathbf{w}_w \in \mathbb{R}^M$  (with  $w = 1, \dots, W$ ) collected into the matrix  $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_W] \in \mathbb{R}^{M \times W}$ . To determine the ROM solution  $\mathbf{v}_R$ , we enforce the orthogonality between the residual  $\mathbf{v} - \mathbf{v}_R$  and the subspace  $\text{span}\{\mathbf{w}_1, \dots, \mathbf{w}_W\}$ ; in other words, we perform a Galerkin projection onto the subspace  $\text{span}\{\mathbf{w}_1, \dots, \mathbf{w}_W\}$ . The governing equation of the ROM then becomes

$$\begin{cases} \mathbf{M}_R \ddot{\mathbf{v}}_R + \mathbf{K}_R(g) \mathbf{v}_R = \mathbf{f}_R(\boldsymbol{\eta}), & t \in (0, T) \\ \mathbf{v}_R(0) = \mathbf{W}^T \mathbf{v}_0(\boldsymbol{\eta}) \\ \dot{\mathbf{v}}_R(0) = \mathbf{W}^T \dot{\mathbf{v}}_0(\boldsymbol{\eta}) \end{cases} \quad (4)$$

where

$$\mathbf{M}_R = \mathbf{W}^T \mathbf{M} \mathbf{W}, \quad \mathbf{K}_R(g) = \mathbf{W}^T \mathbf{K}(g) \mathbf{W}, \quad \mathbf{f}_R = \mathbf{W}^T \mathbf{f}.$$

Eq. (4) is integrated in time to obtain  $\mathbf{V}_R = [\mathbf{v}_{R1}, \dots, \mathbf{v}_{RL}] \in \mathbb{R}^{W \times L}$ , and then projected back onto the original FOM space to obtain the whole solution  $\mathbf{V} \approx \mathbf{W}\mathbf{V}_R$ .

If the FOM arrays in Eq. (2) exhibit an affine parametric dependency, it is possible to write

$$\mathbf{K}(g) = \sum_{p=1}^{P_k} \psi_p(g) \mathbf{K}_p, \quad \mathbf{f}(\boldsymbol{\eta}) = \sum_{p'=1}^{P_f} \psi'_{p'}(\boldsymbol{\eta}) \mathbf{f}_{p'},$$

where  $\psi_p(g) : \{0, \dots, G\} \rightarrow \mathbb{R}$  (with  $p = 1, \dots, P_k$ ) and  $\psi'_{p'}(\boldsymbol{\eta}) : \mathbb{H} \rightarrow \mathbb{R}$  (with  $p' = 1, \dots, P_f$ ) are two sets of scalar functions;  $\mathbf{K}_p \in \mathbb{R}^{M \times M}$  (with  $p = 1, \dots, P_k$ ) is a set of  $g$ -independent matrices;  $\mathbf{f}_{p'} \in \mathbb{R}^M$  (with  $p' = 1, \dots, P_f$ ) is a set of  $\boldsymbol{\eta}$ -independent vectors. For the case at hand, affine parametric dependency is built-in in the formulation of the FOM, since parameters governing both the operational conditions and the damage states are taken as constant in time, and piecewise constants over different spatial subdomains, thus yielding the possibility to factor them out of the assembled matrices. Nonaffine parametric dependency, on the other hand, would require the use of suitable *hyper-reduction* techniques, to restore an approximate affine parametric dependency, see, e.g. [21].

Under the assumption of affine parametric dependency, assembling the ROM arrays in Eq. (4) can be made independent of the FOM dimension  $M$  for any  $\{g, \boldsymbol{\eta}\}$ . Indeed, we have

$$\begin{aligned} \mathbf{K}_R(g) &= \sum_{p=1}^{P_k} \psi_p(g) \mathbf{W}^T \mathbf{K}_p \mathbf{W} = \sum_{p=1}^{P_k} \psi_p(g) \mathbf{K}_R^p, \\ \mathbf{f}_R(\boldsymbol{\eta}) &= \sum_{p'=1}^{P_f} \psi'_{p'}(\boldsymbol{\eta}) \mathbf{W}^T \mathbf{f}_{p'} = \sum_{p'=1}^{P_f} \psi'_{p'}(\boldsymbol{\eta}) \mathbf{f}_R^{p'}, \end{aligned}$$

given that

$$\mathbf{K}_R^p = \mathbf{W}^T \mathbf{K}_p \mathbf{W} \in \mathbb{R}^{W \times W}, \quad \mathbf{f}_R^{p'} = \mathbf{W}^T \mathbf{f}_{p'} \in \mathbb{R}^W,$$

can be computed and stored once for all, with  $\mathbf{K}_R(g)$ ,  $\mathbf{f}_R(\boldsymbol{\eta})$  constructed without repeating the costly assembling operations required by the FE model.

To set  $\mathbf{W}$ , a Proper Orthogonal Decomposition (POD) is performed on the matrix  $\mathcal{S} = [\mathbf{v}_1, \dots, \mathbf{v}_S] \in \mathbb{R}^{M \times S}$  collecting  $S$  snapshots of the FOM. The collected snapshots must embody both the dependence on  $\{g, \boldsymbol{\eta}\}$  and on  $t$ ; details on basis construction in elastodynamics are reported in Algorithm 1. The total number of snapshots collected is  $S = Y \times X$ , where  $Y \geq 1 + G$  is the number of samples of  $\{g, \boldsymbol{\eta}\}$ , so that each damage state is sampled at least once, and  $X$  is the number of samples in time. In order to guide the choice of  $Y$  and  $X$ , we refer, *e.g.* to [6]: here, we only remark that the constraint  $X \leq L$  must be always satisfied. Indeed, to speedup the ROM construction there is often the possibility to restrict the snapshot collection to a small portion of the time window of interest, given that enough information on the dynamic evolution of the system is still captured.

---

**Algorithm 1** POD bases determination in elasto-dynamics, as proposed in [25].

---

- 1: Sample  $\boldsymbol{\eta}_1$  via Latin Hypercube Sampling
  - 2: Collect  $\mathcal{S}_1 = [\mathbf{v}(g_1, \boldsymbol{\eta}_1, t_1) | \dots | \mathbf{v}(g_1, \boldsymbol{\eta}_1, t_X)]$
  - 3:  $\mathbf{W} = \mathbf{POD}(\mathcal{S}_1)$
  - 4: FOR  $\tau = 2, \dots, Y$  DO
  - 5:     Sample  $\boldsymbol{\eta}_\tau$  via Latin Hypercube Sampling
  - 6:     Collect  $\mathcal{S}_\tau = [\mathbf{v}(g_\tau, \boldsymbol{\eta}_\tau, t_1) | \dots | \mathbf{v}(g_\tau, \boldsymbol{\eta}_\tau, t_X)]$
  - 7:      $\mathbf{W}_\tau = \mathbf{POD}(\mathcal{S}_\tau)$
  - 8:      $\mathcal{S} = [\mathbf{W} | \mathbf{W}_\tau]$
  - 9:      $\mathbf{W} = \mathbf{POD}(\mathcal{S})$
  - 10: END FOR
- 

POD of matrix  $\mathcal{S} \in \mathbb{R}^{M \times S}$  is performed via a singular value decomposition according to

$$\mathcal{S} = \mathbf{P}\boldsymbol{\Sigma}\mathbf{Z}^T \quad (5)$$

where:  $\mathbf{P} = [\mathbf{w}_1, \dots, \mathbf{w}_M] \in \mathbb{R}^{M \times M}$  is an orthogonal matrix whose columns are the left singular vectors of  $\mathcal{S}$ ;  $\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_S] \in \mathbb{R}^{S \times S}$  is an orthogonal matrix whose columns are the right singular vectors of  $\mathcal{S}$ ;  $\boldsymbol{\Sigma} \in \mathbb{R}^{M \times S}$  collects the singular values of  $\mathcal{S}$ . When  $M > S$ ,

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_r \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix},$$

where  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq 0$  and  $r$  is the rank of  $\mathcal{S}$ .

The POD bases  $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_W]$  are then obtained by retaining the first  $W \leq S$  left singular vectors in  $\mathbf{P}$ . Among all possible approximations of  $\mathcal{S}$  of rank  $W$ ,  $\mathbf{W}$  captures as much energy of  $\mathcal{S}$  as possible [21, 41]. The normalized reconstruction error  $\varepsilon$  obtained by retaining the first  $W$  modes can be related to the discarded singular values as

$$\varepsilon = \sqrt{\frac{\sum_{s=W+1}^S \sigma_s^2}{\sum_{s=1}^S \sigma_s^2}}, \quad (6)$$

By prescribing a tolerance  $\varepsilon_{tol}$ , such that  $\varepsilon < \varepsilon_{tol}$ , we can automatically set the dimension  $W$  of the ROM.

## 2.4 Fully Convolutional Networks

Once  $\mathbf{D}$  has been constructed according to Eq. (1),  $I_{tr}$  instances are used to train the classifier  $\mathcal{G}$ . First,  $\mathcal{G}$  performs a series of (nonlinear) mappings through basis functions ruled by tunable weights  $\boldsymbol{\Omega}$  [30], whose overall effect is to make the damage classes linearly separable [42]. In this way, a linear mapping, ruled by a weight matrix  $\boldsymbol{\Theta}$  and followed by a softmax function, proves sufficient to perform the classification task. In details, denoting by  $\boldsymbol{\vartheta} = (\vartheta_0, \dots, \vartheta_G)^T \in \mathbb{R}^{G+1}$  the outcome of the linear mapping, the softmax function computes a vector  $\boldsymbol{\varrho} = (\varrho_0, \dots, \varrho_G)^T \in [0, 1]^{G+1}$ , with

$$\varrho_g = \frac{e^{\vartheta_g}}{\sum_{j=0}^G e^{\vartheta_j}} \quad g = 0, \dots, G, \quad (7)$$

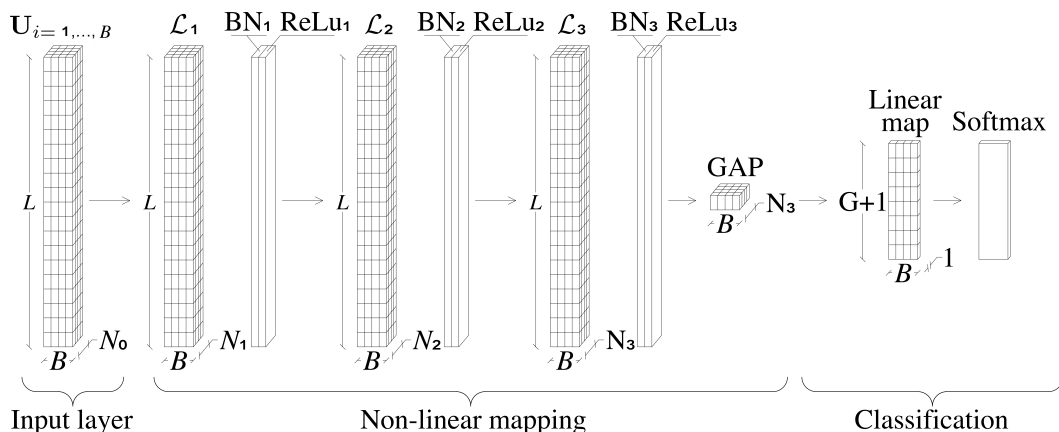
denoting the probability by which the input  $\mathbf{U}$  is assigned to the  $g$ -th damage class. The classification task is then performed by selecting the class  $g$  corresponding to the highest  $\varrho_g$  value.

During the training stage, the classification error is quantified by a loss function  $C$ , assumed to be the cross entropy

$$C(\mathbf{z}, \boldsymbol{\varrho}) = - \sum_{g=0}^G z_g \log(\varrho_g), \quad (8)$$

where  $z_g \in \{0, 1\}$  is the confidence with which the  $g$ -th damage class should be assigned to  $\mathbf{U}$ ;  $\mathbf{z}$  is the vector that collects the confidence values. The training consists in minimizing  $C$  by tuning  $\boldsymbol{\Omega}$  and  $\boldsymbol{\Theta}$  through an iterative procedure; Adam [43], a first-order stochastic gradient descend algorithm, is employed with this aim. At each iteration, a certain number of instances, called mini-batch, are analyzed simultaneously. In the forthcoming example sections, we have employed mini-batches containing  $B = 16$  instances, but in the following we assume that each mini-batch counts just one of them, in order to simplify the notation. At the end of the training, the  $I_{tr}$  instances are processed a number of times named *epochs*.

The nonlinear mappings ruled by  $\boldsymbol{\Omega}$  and the final classification ruled by  $\boldsymbol{\Theta}$  have been performed employing a NN called Fully Convolutional Network (FCN), resembling the one proposed in [23]. The chosen NN architecture, depicted in Fig. 1, can analyze multivariate time series, so that each channel  $\mathbf{u}_n$  is not treated separately, and correlations between different channels can be exploited to improve the classifier effectiveness. The functioning of each block depicted in Fig. 1 is detailed in the following.



**Figure 1:** FCN architecture.

A NN is a computational algorithm that assembles basic units called neurons [42]. Each neuron computes a scalar output  $y$  by operating first a linear mapping of the input  $\mathbf{u} \in \mathbb{R}^\lambda$  (which reads  $\mathbf{u}_n$  for the first layer of the NN, see below) through a weight vector  $\boldsymbol{\omega} \in \mathbb{R}^\lambda$  and a bias term  $\beta$ , and by using next a nonlinear activation function  $\zeta$  according to

$$y = \zeta(\boldsymbol{\omega} \cdot \mathbf{u} + \beta).$$

Typical choices for the activation function are the hyperbolic tangent or the Rectified Linear Unit (ReLU). Thanks to the use of activation functions, NNs can perform tasks that go beyond the ones accomplished by a sequence of linear projections [30]. If  $\lambda$  neurons simultaneously deal with  $\mathbf{u}$ , the output of the transformation is a vector  $\mathbf{y} \in \mathbb{R}^\lambda$ : this collection of  $\lambda$  neurons is called layer. In a DL framework, more layers are usually stacked in order



to form a deep architecture. Several layer typologies exist, differing in terms of the way in which the neurons are connected to the inputs: the layer typology described so far is called fully-connected.

In our NN architecture, convolutional layers are exploited. Convolutional layers are widespread in computer vision [44] and they are gaining attention in signal processing too [45, 46]. In particular, three convolutional layers  $\mathcal{L}_k$  (with  $k = 1, 2, 3$ ) have been employed, each one together with a Batch Normalization (BN)  $\mathcal{B}_k$  and a ReLU activation  $\mathcal{R}_k$ . We call convolutional block the computational unit made up by  $\mathcal{L}_k$ ,  $\mathcal{B}_k$  and  $\mathcal{R}_k$ . By putting three convolutional blocks in sequence and by applying a Global Average Pooling (GAP) [47] to their output, the first part of the NN is constructed.

We now detail how the convolutional units and the GAP work. The adopted notation holds for the first convolutional block and, for this reason, the input is denoted by  $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_{N_0}]$ ; the same reasoning holds for the second and third convolutional blocks, for which the inputs are the outputs of the first ( $\bar{\mathbf{Y}}^1 \in \mathbb{R}^{L \times N_1}$ ) and second convolutional blocks ( $\bar{\mathbf{Y}}^2 \in \mathbb{R}^{L \times N_2}$ ), respectively.

The output  $\mathbf{Y}^1 = [\mathbf{y}_1^1, \dots, \mathbf{y}_{N_1}^1] \in \mathbb{R}^{L \times N_1}$  of  $\mathcal{L}_1$  is computed as

$$\mathbf{y}_b^1 = \sum_{n=1}^{N_0} \omega_b^{1n} * \mathbf{u}_n \quad b = 1, \dots, N_1, \quad (9)$$

where:  $*$ :  $(\mathbb{R}^{H_1} \times \mathbb{R}^L) \rightarrow \mathbb{R}^L$  is the discrete convolution operator [48];  $\Omega_b^1 = [\omega_b^{11}, \dots, \omega_b^{1N_0}] \in \mathbb{R}^{H_1 \times N_0}$  are the weights, called filter kernels, applied to  $\mathbf{u}_n$ ;  $\Omega^1 = [\Omega_1^1, \dots, \Omega_{N_1}^1] \in \mathbb{R}^{H_1 \times N_0 \times N_1}$  is the overall weights set of  $\mathcal{L}_1$ . Bias terms are omitted to simplify the notation.

As a second step, the BN zero-centers and normalizes  $\mathbf{y}_b^1$ , in order to address the issue of the vanishing/exploding gradient [49] that frequently affects the training of NNs in DL.

Finally, the adopted ReLU activation function reads

$$\bar{y}_b^{1l} = \mathcal{R}_1(\mathcal{B}_1(y_b^{1l})) = \max(0, \mathcal{B}_1(y_b^{1l})) \quad l = 1, \dots, L, \quad (10)$$

where  $y_b^{1l}$  is the  $l$ -th entry of  $\mathbf{y}_b^1$ .

Both the BN and the ReLU activation function do not involve any tunable parameter, so that the outcome of the first part of the NN is ruled just by  $\Omega = [\Omega^1 | \Omega^2 | \Omega^3]$ , where  $\Omega^1$ ,  $\Omega^2$  and  $\Omega^3$  are the weights employed by  $\mathcal{L}_1$ ,  $\mathcal{L}_2$  and  $\mathcal{L}_3$ , respectively. Each channel  $\bar{\mathbf{y}}_b^3$ , output of the third computational block, is still shaped as a time series of length  $L$ , but does not represent a displacement time history anymore, as it becomes a feature of  $\mathbf{U}$ . A feature is an optimized representation of  $\mathbf{U}$ , optimality being meant in the sense of the minimization of the loss function  $C$ . Also the GAP does not use any tunable parameter: the input  $\bar{\mathbf{Y}}^3 = [\bar{\mathbf{y}}_1^3, \dots, \bar{\mathbf{y}}_{N_3}^3] \in \mathbb{R}^{L \times N_3}$  is handled to compute an average value for each channel  $\bar{\mathbf{y}}_b^3$  as

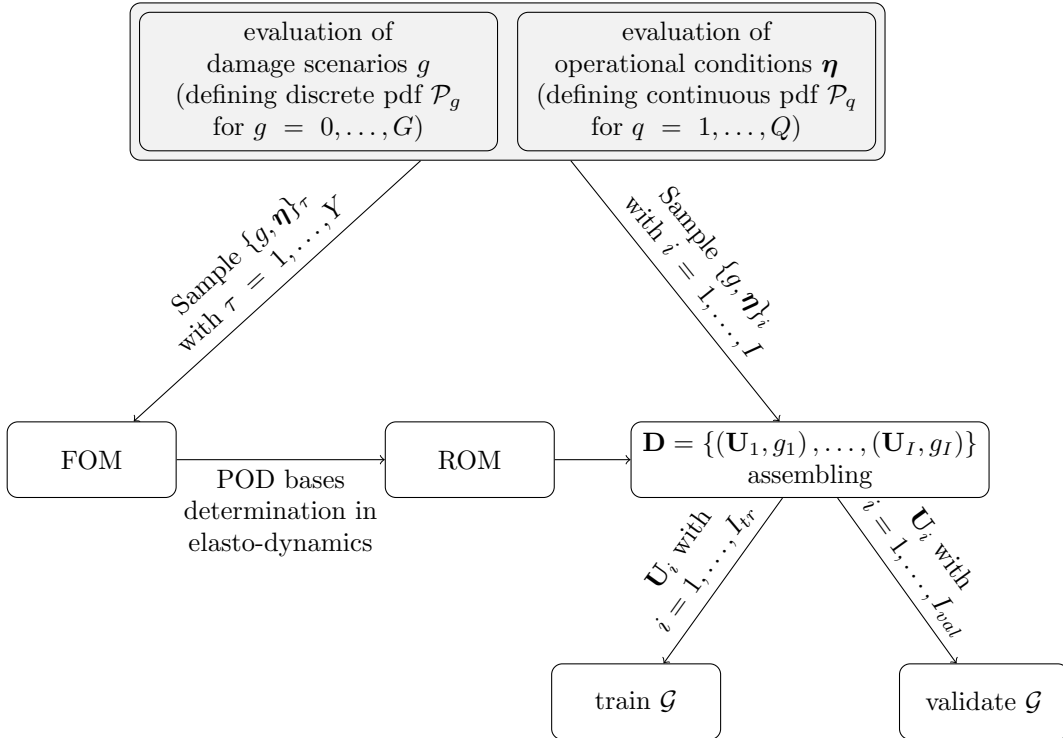
$$\text{GAP}(\bar{\mathbf{y}}_b^3) = \frac{1}{L} \sum_{l=1}^L \bar{y}_b^{3l} \quad b = 1, \dots, N_3. \quad (11)$$

The GAP outcomes are synthetic description of the channel contents, highly informative for the classification task.

The implementation of the FCN architecture has been developed making use of the Keras API, based on Tensorflow. The hyperparameters featured by the NN (*e.g.* the number of filters  $N_k$ ) and controlling the training (*e.g.* the number of epochs) have been initially set according to [23, 20]. A further hyperparameter tuning has been carried out through the repeated evaluation of the classification accuracy of  $\mathcal{G}$  on the case studies discussed next. Specific attention has been paid to avoid overfitting of the training data: in Tab. 1 the hyperparameters values employed for the two case studies are reported.

	Portal frame	Integral bridge
$N_0$	6	4
$N_1, N_2, N_3$	16, 32, 16	16, 32, 16
$H_1, H_2, H_3$	8, 5, 3	8, 5, 3
$B$	16	16
$I$	10,000	10,000
n° epochs	500	1000

**Table 1:** FCN hyperparameters.



**Figure 2:** Methodology flowchart.

In this section, we have discussed the elements constituting the proposed methodology: the definition of a stochastically parametrized FOM of the monitored structure; the determination of a ROM keeping track of the system parametric and temporal dependence; the construction of a dataset  $\mathbf{D}$ ; the training of a FCN-based classifier  $\mathcal{G}$ . To further clarify the connection between these different steps, Fig. 2 is reported.

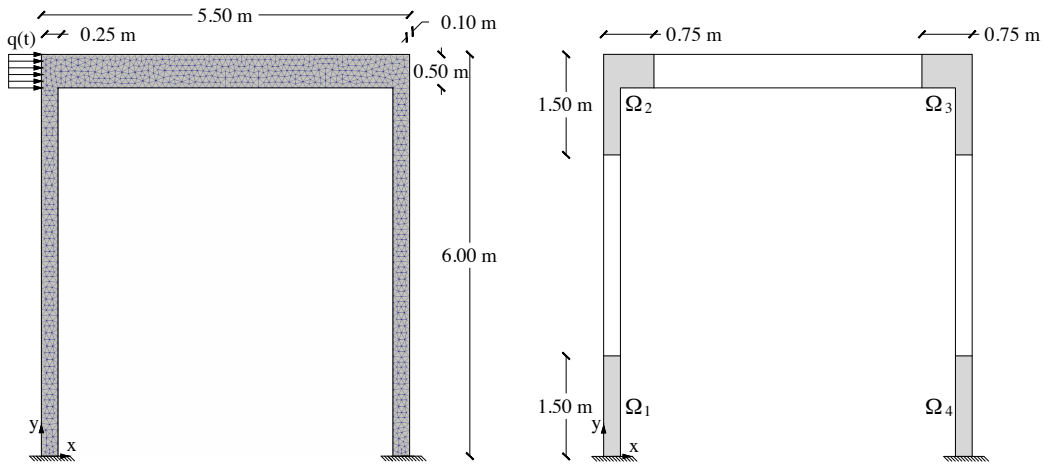
### 3 Case study 1 - Portal frame

The proposed methodology has been first used for a two-dimensional, single-storey frame subjected to a dynamic load. The purpose of this analysis is to verify the impact on the classifier performance of the ROM handling a variable damage level and of different values of the Signal to Noise Ratio (SNR) characterizing the sensor accuracy.

### 3.1 Portal frame - FOM

The two-dimensional portal frame depicted in Fig. 3 has been numerically modeled via a FE discretization consisting of 1882 constant strain triangles (CSTs) and 1884 dofs. Time discretization has been performed by partitioning  $(0, T)$  into subintervals of size  $5 \cdot 10^{-3}$  s. The structural thickness has been assumed to be 0.1 m, so that a plane stress condition has been adopted. The structure has been assumed to be made of concrete, whose mechanical properties are: Young's modulus  $E = 30$  GPa, Poisson's ratio  $\nu = 0.2$ , density  $\rho = 2500$  kg/m<sup>3</sup>. The structure has been excited by a distributed load, acting on the left column in correspondence of the deck, varying in time as  $q(t) = A |\sin(2\pi ft)|$ . The load amplitude  $A$  and the frequency  $f$  of the load have been modeled as random variables, with uniform pdfs  $\mathcal{U}_A(10, 50)$  kPa and  $\mathcal{U}_f(50, 95)$  Hz.

In Fig. 3, the considered damage scenarios are shown on the right. Each relevant structural state  $g \in \{1, 2, 3, 4\}$  is linked to a damage of the corresponding subdomain  $\Omega_1, \dots, \Omega_4$ , while the undamaged scenario is given by  $g = 0$ . If not stated otherwise, damaged and undamaged scenarios have been assumed to have equal probability to be encountered during the monitoring stage, therefore a discrete uniform pdf  $\mathcal{U}_g(0, \dots, 4)$  has been assumed for  $g$ . The damage level  $\delta$ , which represents the stiffness reduction applied to the considered subdomain, has been modeled as a continuous random variable with uniform pdf  $\mathcal{U}_\delta(2\%, 25\%)$ . Therefore, for the current analysis the parametric dependence has involved  $\boldsymbol{\eta} = \{A, f, \delta\}$ . As load amplitude  $A$ , load frequency  $f$  and damage level  $\delta$  may vary continuously, the adopted pdfs  $\mathcal{U}_A$ ,  $\mathcal{U}_f$  and  $\mathcal{U}_\delta$  are continuous too; on the other hand, since the damage scenario can only take values in the discrete set  $\{0, 1, 2, 3, 4\}$ , a discrete pdf  $\mathcal{U}_g$  has been adopted.

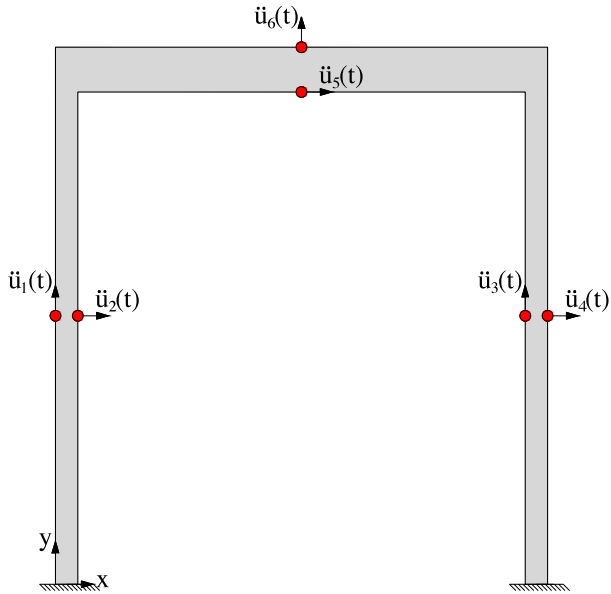


**Figure 3:** Portal frame: (left) loading and space discretization; (right) considered damage scenarios in the subdomains  $\Omega_1, \dots, \Omega_4$ .

The dataset  $\mathbf{D}$  has been built by collecting instances, together with the corresponding labels, obtained for parameters sampled via LHS from the parametric space spanned by  $g$  and  $\boldsymbol{\eta}$ . In Tab. 2 the first eight vibration frequencies and relevant periods of vibration of the model are listed. The monitoring system consists of  $N_0 = 6$  sensors, recording either the horizontal or vertical accelerations as depicted in Fig. 4. The signals have been recorded with a sampling frequency of 200 Hz, allowing to properly account for the first seven structural frequencies without incurring in aliasing. Each numerical simulation covers 1 s of duration: therefore, in the monitoring interval, each record includes  $L = 200$  samples.

Mode	$f_{Num}$ [Hz]	Period [s]
1	4.02	0.2488
2	24.18	0.0413
3	31.31	0.0319
4	36.71	0.0272
5	79.44	0.0126
6	80.89	0.0124
7	96.70	0.0103
8	128.66	$7.77 \cdot 10^{-3}$

**Table 2:** Portal frame: structural frequencies.



**Figure 4:** Portal frame: sensor system arrangement.

### 3.2 Portal frame - ROM

To build the ROM, the snapshots have been collected for different values of  $\{g, \boldsymbol{\eta}\}$ . At this stage, no noise has been added to corrupt the model outcomes. The number of samples of  $\{g, \boldsymbol{\eta}\}$  has been fixed to  $Y = 200$ ; the number of samples in time has been instead fixed to  $X = 100$  for time windows of 0.5 s. The total number of collected snapshots therefore amounts to  $S = 200 \times 100 = 20000$ .

Including the damage level  $\delta$  inside  $\boldsymbol{\eta}$  proved necessary in order to identify the presence of either a minimal or a moderate structural damage. Indeed, it has been observed that a classifier trained for a fixed damage level does not work properly in recognizing structural states characterized by a different level of damage. As an example, Tab. 3 provides the performance, given in terms of classification accuracy, of a classifier  $\mathcal{G}$  trained for  $\delta = 25\%$  in recognizing instances characterized by different values of  $\delta$ . The performance drops while moving away from the training value, due to major difficulties in recognizing a lower damage level, and to the fact that the NN has only been trained to distinguish between the conditions characterized by either  $\delta = 25\%$  or  $\delta = 0\%$ .

The damage level  $\delta$  also shows a close relationship with the number  $W$  of POD bases: Tab. 4 shows how  $W$  tends to reduce, as  $\delta$  decreases. This happens because the smaller the ROM dimension  $W$  by which the prescribed error tolerance  $\varepsilon = 10^{-4}$  is achieved, the more similar are the scenarios to be described. This also means that the ROM is much more prone to fail when modelling structural states characterized by smaller value of  $\delta$ , because a smaller number of POD bases hardly catches the effect of small damages, as confirmed by the classification performance obtained with different classifiers trained and tested for an assigned value of  $\delta$ , see Tab. 5. Even if not reported here for the sake of brevity, such accuracy has turned out to be not affected by the structural damping: the same analyses have been run by allowing for a Rayleigh damping featuring a ratio of 5% for the first two structural modes, with no variations with respect to the values reported in Tab. 5.

Additional results have been obtained by considering the discrete pdf  $\mathcal{U}_g$  relevant to the damage scenarios to be not uniform. This may be traced back to a former inspection of the structure, to ascertain if a specific damage state can occur more likely than others

(maybe due to some defects in the initial state), or to a global sensitivity analysis to provide insights into the links between input loading and probability to incept a specific damage pattern. Datasets can be generated handling different  $\mathcal{U}_g$ ; results are here discussed for a case featuring a probability of damage scenario  $g = 2$  to occur, twice the others. The load amplitude and frequency have been instead extracted from the same pdfs  $\mathcal{U}_A$  and  $\mathcal{U}_f$  defined before. Maps of the sampled values are reported in Fig. 5, in terms of projections onto the planes  $A - f$ ,  $A - g$  and  $f - g$ , where it can be easily recognized that the damage scenario  $g = 2$  has a higher probability testified by the denser distribution of the samples at varying magnitude and frequency of the load. For  $\delta = 25\%$ , the accuracy in classification relevant to case (a) has been the already considered 100% reported in Tab. 5; the same accuracy has been obtained for case (b). This outcome testifies that the proposed method is robust against improper assumptions regarding the probability of the different damage scenarios to occur.

$\delta$ [%]	Accuracy [%]
25	100
20	90
15	82
10	36
5	22
2	22

**Table 3:** Portal frame: accuracy performance obtained with a classifier trained for  $\delta = 25\%$ , in recognizing structural states characterized by different values of the damage level  $\delta$ .

$\delta$ [%]	POD bases
25	65
20	64
15	58
10	56
5	53
2	47

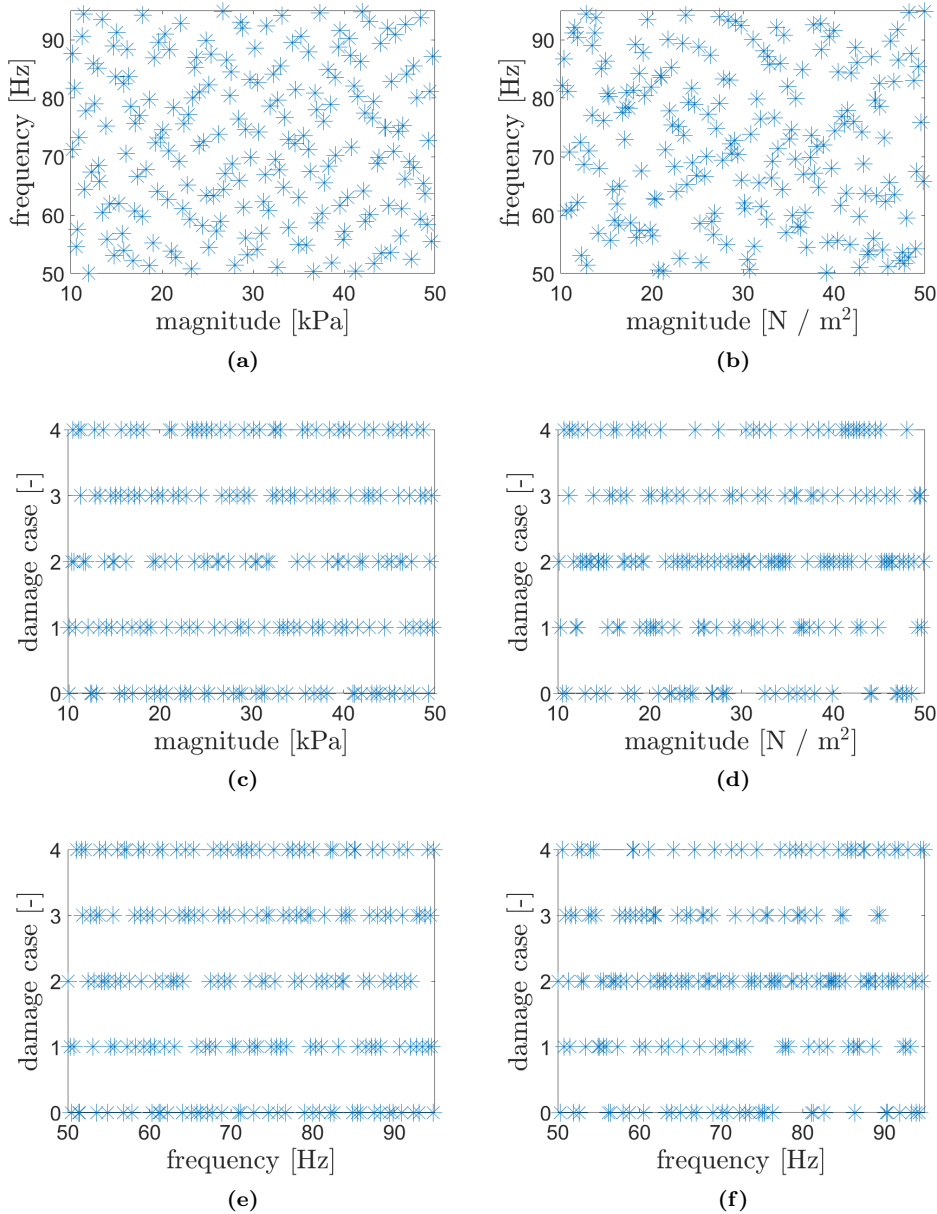
**Table 4:** Portal frame: number  $W$  of POD bases at varying damage level  $\delta$ .

$\delta$ [%]	Accuracy [%]
25	100
20	100
15	96
10	92
5	84
2	62

**Table 5:** Portal frame: accuracy performance obtained with classifiers trained and tested for several values of the damage level  $\delta$ .

The ROM has been built according to the procedure described in Algorithm (1). In Figs. 6-7, the normalized singular values  $\sigma_s^T/\sigma_1^T$  and  $\sigma_s^{g\eta}/\sigma_1^{g\eta}$ , with their typical descendant behavior, are reported for the POD in time and over the parametric space, respectively. The dashed horizontal line in the graph refers to the last selected POD basis, thus the correspondent abscissa identifies the overall number of selected POD bases ensuring a reconstruction error below the prescribed error tolerance  $\varepsilon < \varepsilon_{tol} = 10^{-4}$ . An excellent approximation capacity has been achieved by relying upon 59 POD bases only, instead of the original 1884 dofs. In Fig. 8 the first 8 POD bases are reported: higher order POD bases feature more complex shapes, useful to simulate the effect of localized damages. Due to the mentioned reduction in the number of the dofs from the FOM to the ROM, the computing time required by each simulation has decreased from 45 s to 1.5 s, with a speedup of 30 (computations have been run on a PC featuring an Intel (R) Core™, i5 CPU @ 2.6 GHz and 8 GB RAM).

Since acceleration measurements have been exploited for SHM purposes, we have kept  $\varepsilon_{tol}$  small also considering that  $\varepsilon$  has been based on the reconstruction of the displacement field. As an example, in Fig. 9 the FOM and ROM time histories are compared for the (unobserved) horizontal acceleration at the top-right corner of the frame, for different values of  $\varepsilon_{tol}$  and for  $g = 0$  (undamaged state),  $A = 30$  kPa,  $f = 80$  Hz,  $\delta = 0$ . The ROM solution relevant to  $\varepsilon_{tol} = 10^{-3}$  has turned out to be excessively inaccurate, so that  $\varepsilon_{tol} = 10^{-4}$  has been adopted for the construction of **D**.

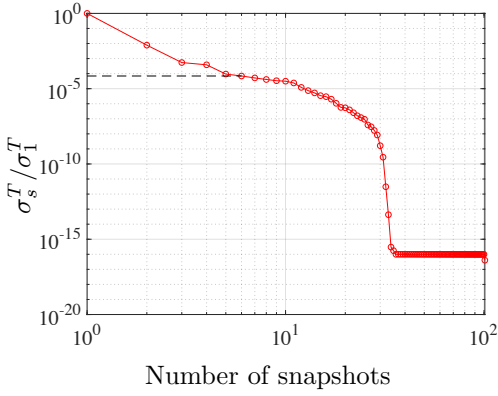


**Figure 5:** Portal frame: sampled values of load amplitude  $A$ , load frequency  $f$  and damage scenario  $g$  in: (left column) case (a), featuring a uniform pdf  $\mathcal{U}_g$  over all the damage scenarios; (right column) case (b), featuring a probability of scenario  $g = 2$  to occur twice the others.

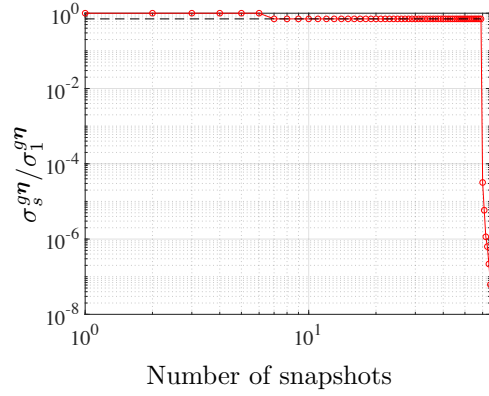
### 3.3 Portal frame - Classification outcomes

According to the NN hyperparameters setting discussed in Sec. 2.4,  $\mathcal{G}$  has been trained and validated on  $I_{tr} + I_{val} = 10,000$  instances with ratio 75 : 25, for 500 epochs. A further discussion on the employed number of filters  $N_1$ ,  $N_2$  and  $N_3$  will be provided in the following.

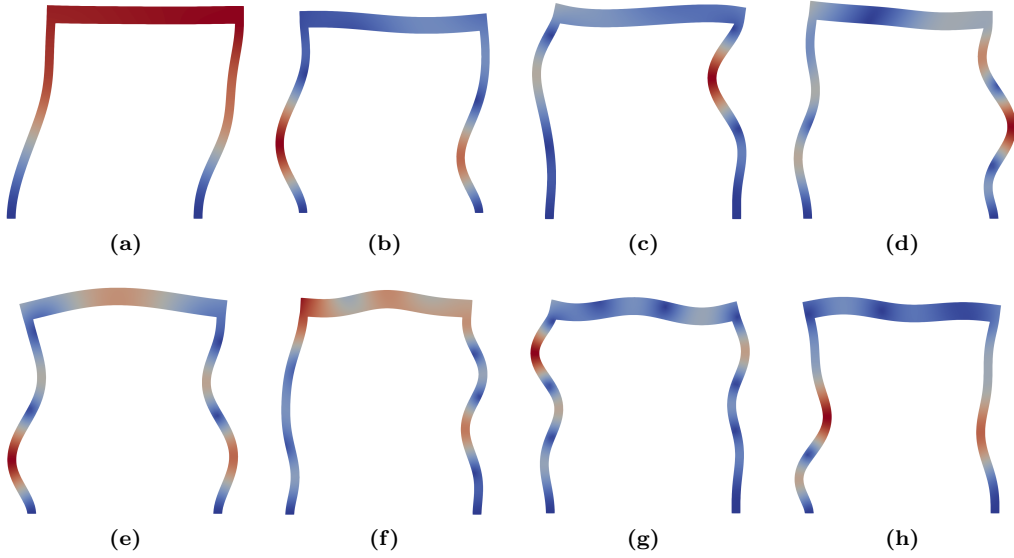
For the noise-free case, Figs. 10 and 11 show the evolutions of the loss and of the accuracy during training, respectively. The iteration number, determined by the number of epochs and by the dimension of the mini-batches, accounts for the number of times the NN weights are modified during the training process. As expected, the first portion of training shows



**Figure 6:** Portal frame: POD in time. Descent of the singular values  $\sigma_s^T$  normalized with respect to  $\sigma_1^T$ .



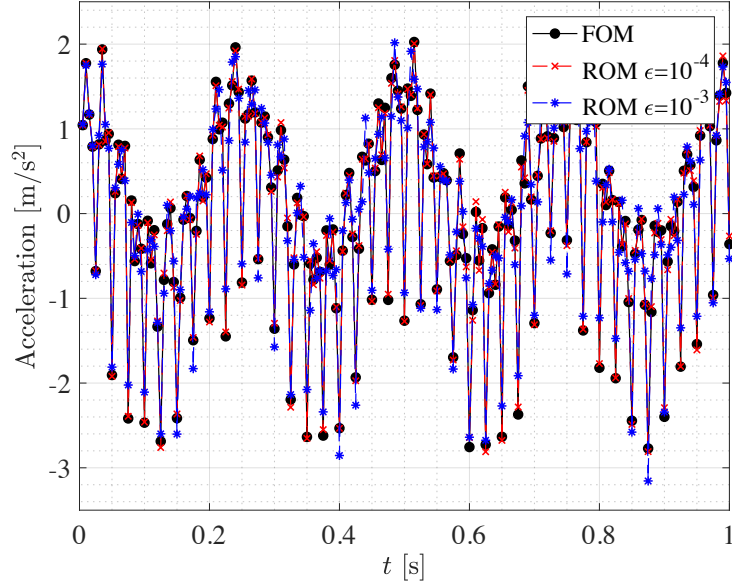
**Figure 7:** Portal frame: POD over parameters. Descent of the singular values  $\sigma_s^{g\eta}$  normalised with respect to  $\sigma_1^{g\eta}$ .



**Figure 8:** Portal frame: POD bases.

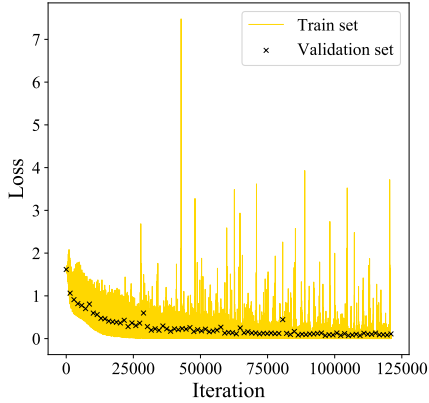
the highest gains in terms of classification accuracy. The spikes, observable both in the loss and accuracy graphs, are due to the different classification performances obtained on different mini-batches. At the end of the training, we have obtained classification accuracies of 98.91% and 97.68%, respectively for the training and validation sets.

The generalization capabilities of  $\mathcal{G}$  have been evaluated against a test set made of 200 pseudo-experimental instances generated through the FOM. The classification test results are summarized by the confusion matrix in Fig. 12. Even for this test, no noise effects have been taken into account. The classification task has been carried out with a global accuracy of 84.5%, but two different sources of error can be highlighted. First, 40% of test instances featuring the scenario  $g = 2$  have been misclassified as  $g = 1$ ; with a smaller frequency, 22.5%, the same happens for  $g = 4$  and  $g = 3$ . This is due to the similar influence of those damage scenarios on the structural response, by virtue of the structural layout and of the applied load. Second, 15% of test instances featuring  $g = 0$  have been misclassified as well. This is due to the variability of  $\delta$ , as smaller values of  $\delta$  provide the additional

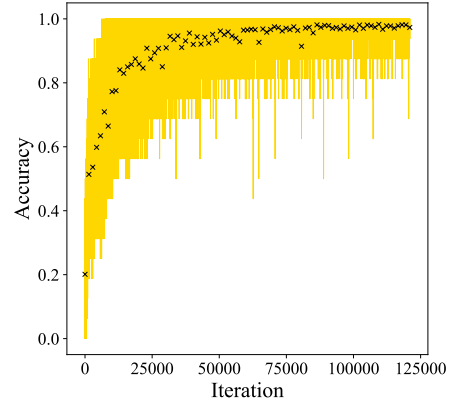


**Figure 9:** Portal frame: comparison between acceleration time histories generated through the FOM and the ROM for different values of  $\varepsilon_{tol}$ .

issues discussed before: an increased difficulty to distinguish a damaged scenario from an undamaged one; a major difficulty for the ROM to properly describe the damaged scenarios. The other way around, none of the damaged scenarios has been misclassified as undamaged. These effects of  $\delta$  have been observed for a relevant number of tests, carried out with the same classifier, by processing different test sets characterized by a fixed damage level  $\delta$ ; corresponding outcomes are summarized in Tab. 6.



**Figure 10:** Portal frame: FCN training. Loss function evolution on the training and validation sets for the noise-free case.



**Figure 11:** Portal frame: FCN training. Accuracy function evolution on the training and validation sets for the noise-free case.



Accuracy: 84.50%

g=0	85.0% 34	0.0% 0	0.0% 0	0.0% 0	0.0% 0
g=1	5.0% 2	100.0% 40	40.0% 16	0.0% 0	0.0% 0
g=2	5.0% 2	0.0% 0	60.0% 24	0.0% 0	0.0% 0
g=3	0.0% 0	0.0% 0	0.0% 0	100.0% 40	22.5% 9
g=4	5.0% 2	0.0% 0	0.0% 0	0.0% 0	77.5% 31
	g=0	g=1	g=2	g=3	g=4

**Figure 12:** Portal frame: testing of the FCN. Confusion matrix for the noise-free case.

$\delta$ [%]	Accuracy [%]
25	66
20	78
15	94
10	94
5	78
2	60

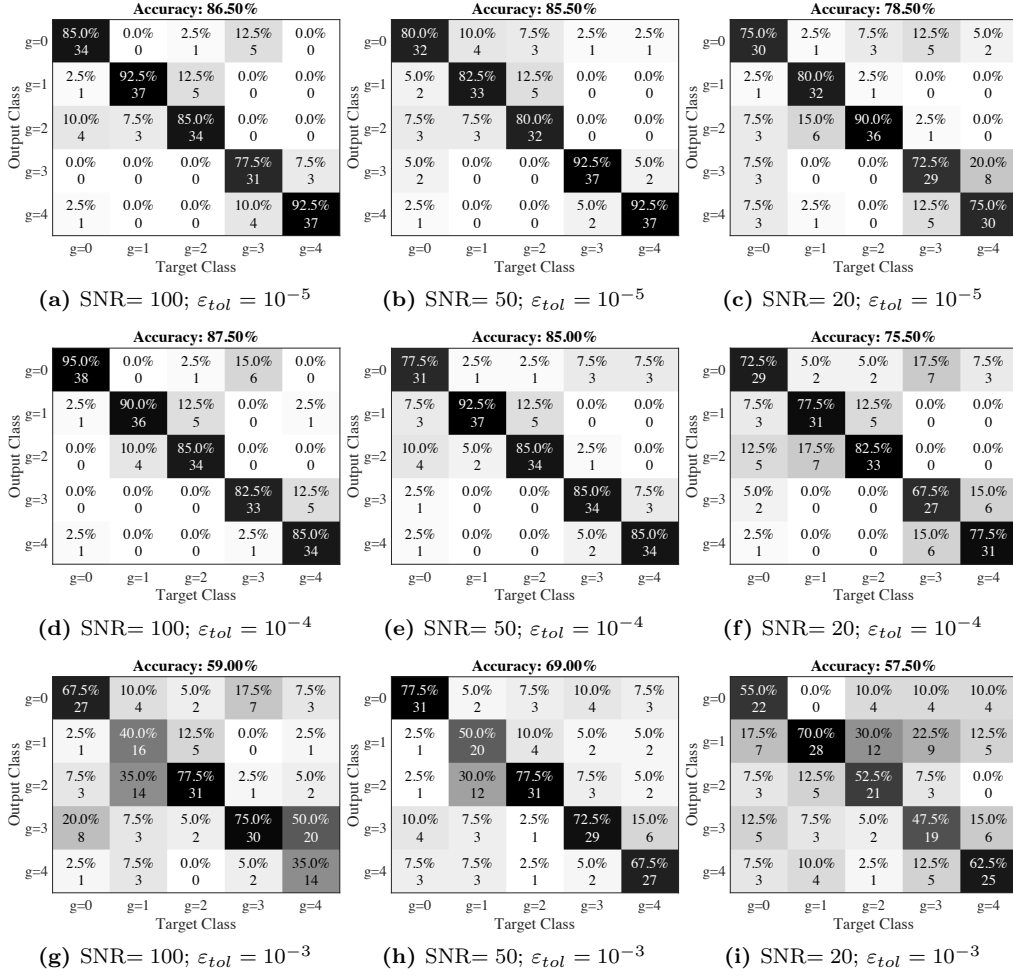
**Table 6:** Portal frame: classification accuracy in recognizing structural states characterized by a fixed damage level  $\delta$  for the noise-free case.

To assess the impact of the noise level on the performance of  $\mathcal{G}$ , the classifier has been next trained and tested on datasets featuring SNR = 100, 50 and 20. The chosen SNR values are representative of the self-noise of accelerometers usually employed in the monitoring of civil structures [50, 51]. A certain SNR, indicator of the noise level affecting the vibration signals, can be obtained by adding a white noise to the synthetic structural recordings. Despite the relative simplicity of this procedure, the white noise allows to accurately mimic the signal perturbation affecting the measurements of real-life sensors, among which also micro-electro mechanical accelerometers [52, 53].

To get a more clear picture on how SNR affects the classification accuracy together with the ROM complexity, three ROMs, each of them featuring a specific  $\varepsilon_{tol} = 10^{-3}$ ,  $10^{-4}$  and  $10^{-5}$ , have been used for the generation of  $\mathbf{D}$ . The classification outcomes are reported in Tab. 7 for two different choices of the NN filters,  $N_1 = 8$ ,  $N_2 = 16$  and  $N_3 = 8$  in Tab. 7a and  $N_1 = 16$ ,  $N_2 = 32$  and  $N_3 = 16$  in Tab. 7b, respectively. The performance of  $\mathcal{G}$  has been evaluated against the aforementioned three FOM test sets, with the results obtained after training the classifier for each one of the nine combinations of  $\varepsilon_{tol}$  and SNR levels. From the results in Tab. 7, it can be observed that both  $\varepsilon_{tol}$  and SNR have a relevant impact on the accuracy of  $\mathcal{G}$ . Lower performances follow lower SNRs, especially when combined with large values of  $\varepsilon_{tol}$ . Nevertheless, by setting  $\varepsilon_{tol}$  small enough (e.g. to  $\varepsilon_{tol} = 10^{-4}$ ), a global accuracy larger than 75% can be always attained. By enhancing the approximation capacity of the ROM, the classifier thus gets more robust against the noise.

A further insight into the effects of  $\varepsilon_{tol}$  and of SNR on the classification accuracy can be gained by looking at the confusion matrices reported in Fig. 13. The two main sources of errors highlighted before, *i.e.* the misclassification between  $g = 1$  and  $g = 2$ , and between  $g = 3$  and  $g = 4$ , are encountered even under different SNR levels. Smaller values of SNR worsen the classification performance, even without showing a clear path to penalize some specific damage scenarios.

Tab. 7 has been reported to clarify the criteria used to set the number of filters  $N_1$ ,  $N_2$  and  $N_3$ . By comparing the classification accuracy on the training and validation sets, the classifier employing  $N_1 = 16$ ,  $N_2 = 32$  and  $N_3 = 16$  seems to slightly overfit the training data. Indeed, there is a disparity between the higher performance on the training set and the lower performance on the validation set. This tendency is even more evident when datasets featuring high SNR values are considered, due to the combined effect of the high number of NN weights and of the greater uninformative content induced by noise. The combination of these factors does not lead the NN to filter noise, because the NN has enough weights to keep memory of it. Similar outcomes are reported for the training and test sets. By lowering



**Figure 13:** Portal frame: FCN testing. Confusion matrices at varying values of  $\varepsilon_{tol}$  and SNR.

the number of filters to  $N_1 = 8$ ,  $N_2 = 16$  and  $N_3 = 8$ , that is by halving the overall number of weights in  $\Omega$ , it is possible to reduce this tendency; on the other hand, the accuracy of the NN is reduced. In light of this, we have chosen an architecture employing  $N_1 = 16$ ,  $N_2 = 32$  and  $N_3 = 16$  to carry out the analysis, even if it slightly overfits the data.

To assess the robustness of the proposed methodology, the classifier  $\mathcal{G}$  has been tested in recognizing structural states characterized by a stiffness reduction in subdomains different from those used to train it, see Fig. 3 and compare it with Fig. 14: the damaged subdomains are approximately half in size of those used to construct  $\mathbf{D}$ . In this case, a noise-free condition has been considered. The pseudo-experimental instances used for testing have been generated via FOM. The obtained results, summarized by the confusion matrix in Fig. 15, confirm the robustness of our methodology. The global accuracy is practically unchanged with respect to the previous case (85.50% vs 84.50%) and, furthermore, the sources of the misclassification error look almost the same. The only remarkable difference is linked to a misclassification of the damaged scenarios as undamaged; this outcome is somehow expected since, as shown in Tab. 8, the damaged subdomains reduced in size do have a smaller impact on the structural response to the given loadings.

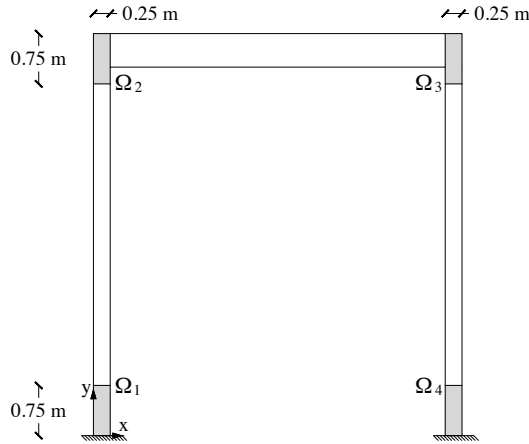
$\varepsilon_{tol} \setminus \text{SNR}$	training set			validation set			test set		
	100	50	20	100	50	20	100	50	20
$10^{-5}$	87.9%	86.3%	79.3%	81.0%	80.7%	72.2%	79.5%	79.5%	71.5%
$10^{-4}$	87.3%	86.6%	80.1%	82.2%	79.8%	72.6%	81.5%	81.5%	68.5%
$10^{-3}$	90.0%	88.0%	84.2%	85.2%	82.2%	76.0%	64.0%	59.5%	62.5%

(a)  $N_1 = 8, N_2 = 16, N_3 = 8$ .

$\varepsilon_{tol} \setminus \text{SNR}$	training set			validation set			test set		
	100	50	20	100	50	20	100	50	20
$10^{-5}$	96.3%	94.5%	92.7%	87.3%	85.0%	78.7%	86.5%	85.5%	78.5%
$10^{-4}$	96.1%	94.9%	93.3%	87.6%	84.2%	79.5%	87.5%	85.0%	75.5%
$10^{-3}$	93.8%	95.9%	95.0%	86.7%	85.1%	79.8%	59.0%	69.0%	57.5%

(b)  $N_1 = 16, N_2 = 32, N_3 = 16$ .

**Table 7:** Classification performance, in terms of global accuracy on the training, validation and test sets, as affected by the values of SNR and  $\varepsilon_{tol}$ .



**Figure 14:** Portal frame: reduced-size damaged regions.

## 4 Case study 2 - Railway bridge

The second case study adopted to assess the performance of the proposed methodology, consists of an integral concrete portal frame railway bridge. Here, the effect of the sensor noise has been disregarded as the focus is on handling the effects of a dynamic moving load. The railway bridge, located along the Bothnia line in the urban area of Hörnefors in the northern Sweden, is depicted in Fig. 16. The bridge has a span of 15.7 m, a free height of 4.7 m, a width of 5.9 m (edge beams excluded) and it does not have any expansion joint or supporting device in between the deck and the abutments. The deck has a thickness of 0.5 m, whilst the frame walls have a thickness of 0.7 m; the wing walls, stretching out in longitudinal direction up to 8 m at the top, have a thickness of 0.8 m. The foundation system consists in a couple of plates connected by two stay beams and supported by pile groups. The bridge superstructure consists of a single ballasted track resting on sleepers spaced 0.65 m apart, while the ballast layer is assumed to have a depth of 0.6 m and a width of 4.3 m.

**Accuracy: 85.50%**

Output Class	g=0	g=1	g=2	g=3	g=4
g=0	87.5% 35	0.0% 0	10.0% 4	0.0% 0	0.0% 0
g=1	0.0% 0	95.0% 38	5.0% 2	0.0% 0	0.0% 0
g=2	2.5% 1	2.5% 1	75.0% 30	0.0% 0	0.0% 0
g=3	10.0% 4	0.0% 0	2.5% 1	72.5% 29	2.5% 1
g=4	0.0% 0	2.5% 1	7.5% 3	27.5% 11	97.5% 39
	g=0	g=1	g=2	g=3	g=4

Target Class

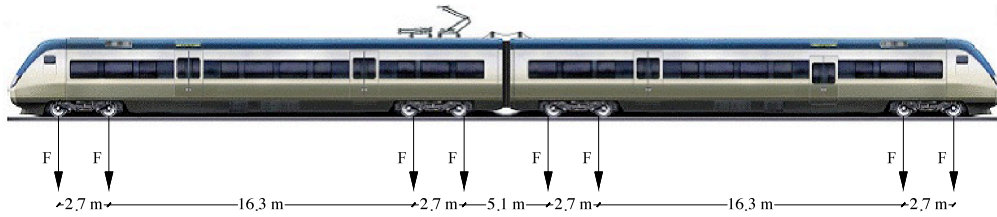
**Figure 15:** Portal frame: FCN testing. Confusion matrix for the reduced damage scenarios.

$\delta$ [%]	Accuracy
25	74
20	84
15	92
10	66
5	56
2	42

**Table 8:** Portal frame: FCN testing. Classification accuracy obtained in recognizing damaged regions reduced in size and characterized by a damage level  $\delta$  in the noise-free case.

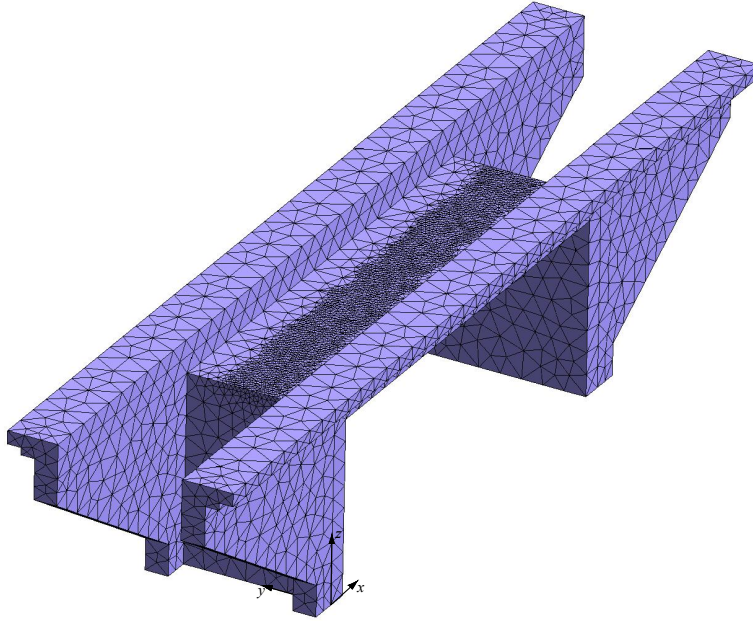


**Figure 16:** Hörnefors railway bridge.



**Figure 17:** Gröna Tåget train type (adapted from [54]).

The structure is loaded by the passage of trains of type Gröna Tåget (Fig. 17), composed of two wagons, in transit with a speed  $\gamma$  ranging between 160 km/h and 215 km/h, and having in total 8 axles. All the geometrical and mechanical data used to model the structure and the loads have been taken from [54, 55], where the relevant soil-structure interaction was studied.



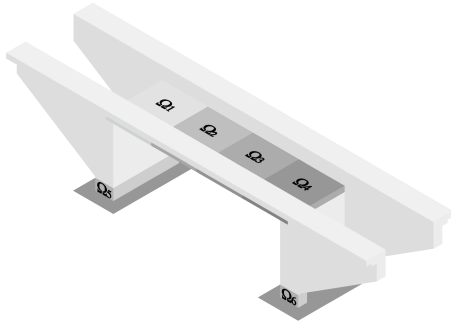
**Figure 18:** FE discretization of the Hörnefors railway bridge.

#### 4.1 Railway bridge - FOM

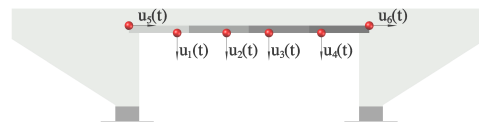
The structure has been discretized with 15,075 four node tetrahedral elements, as shown in Fig. 18, with a total of 15,300 dofs. To properly account for the acting loads, the characteristic size of the elements has been set to 0.15 m for the deck, while it has been set to 0.80 m elsewhere. The bridge has been assumed to be perfectly clamped at the bases. The adopted mechanical properties are those of a concrete class C35/45:  $E = 34$  GPa,  $\nu = 0.2$ ,  $\rho = 2500$  kg/m<sup>3</sup>. The ballast layer, whose density is  $\rho = 1800$  kg/m<sup>3</sup>, has been accounted for by modifying the density of concrete of the deck and the edge beams, thus providing the additional mass resting on the deck. Time discretization has been performed by partitioning  $(0, T)$  with subintervals of size  $2.5 \cdot 10^{-3}$  s, set to account for the maximum train speed of 215 km/h with the element size of 0.15 m. The embankments have been modelled through distributed springs over the lateral surfaces in contact with the ground: this is equivalent to adopting a Robin mixed boundary condition (with elastic coefficient  $a_{Robin} = 10^8$  N/m<sup>3</sup>) in the numerical model.

Six damaged structural states  $g \in \{1, 2, 3, 4, 5, 6\}$ , schematically represented in Fig. 19, have been considered further to the undamaged state  $g = 0$ : the damaged states feature a stiffness reduction in the corresponding subdomain  $\Omega_1, \dots, \Omega_6$ . Each time the mechanical problem has been solved, the damage scenario  $g$  and the damage level  $\delta$  have been respectively sampled via LHS from the uniform discrete pdf  $\mathcal{U}_g(0, \dots, 6)$  and from the uniform continuous pdf  $\mathcal{U}_\delta(5\%, 25\%)$ , respectively. The extreme values of pdf  $\mathcal{U}_\delta$  have been selected in order to assess if small damage events can be detected, distinguished from those characterized by a much larger reduction of the local mechanical properties, and also localized in real-life applications. This approach is obviously intended to work at the structural level, and disregard the features of microcracking patterns in the concrete structure, which are wholly measured through the adopted damage indices. Such an approach was already adopted in the context of SHM in several studies; without any aim to provide an exhaustive account of the literature, readers are referred to, e.g. [56, 57] and also standards like [58].

The monitoring system has been assumed to be composed of  $N_0 = 6$  sensors, placed as shown in Fig. 20, and recording the vertical displacements of the deck and the horizontal displacements at the top of the frame walls. All the signals have been recorded with a sampling frequency of 400 Hz within a monitoring window  $T = 1.5$  s, which allows the train to completely cross the bridge even if traveling at the lowest speed.



**Figure 19:** Railway bridge: considered subdomains  $\Omega_1, \dots, \Omega_6$ . The  $g$ -th damage scenario refers to a localised stiffness reduction in the  $g$ -th subdomain.



**Figure 20:** Railway bridge: monitoring system.

As for the train, the convoy velocity  $\gamma$  and the mass  $\beta$  carried by a single axle (or equivalently the load  $F$  released by the single axle to the rails) have been modelled as random variables with uniform pdf  $\mathcal{U}_\gamma(160, 215)$  km/h and  $\mathcal{U}_\beta(16, 22)$  ton. The load is transmitted from the rails to 25 sleepers that cover the entire deck; the compressive distributed load under the sleepers is then transmitted on its own to the ballast layer with a slope 4 : 1 according to Eurocode 1 [59], so that the loaded surface amounts to  $0.55 \text{ m} \times 2.1 \text{ m}$ . The maximum compressive load value that the train can generate on the sleepers is accordingly  $p_{max} = \frac{F}{(0.55 \cdot 2.1) \text{ m}^2}$ ; the moving load system is finally given as  $P(t, \gamma, x) = \sum_\xi p^\xi(t, \gamma, x)$ , with  $\xi = 1, \dots, 25$ , and [59]

$$p^\xi(t, \gamma, x) = \sum_\alpha p_{max} \cdot A^\xi(x) \cdot A_\alpha^\xi(t, \gamma) \quad \alpha = 1, \dots, 8.$$

where  $A^\xi(x)$  is the space activation function, and  $A_\alpha^\xi(t, \gamma)$  is the time modulation function related to the  $\alpha$ -th axle.

The space activation function  $A^\xi(x)$  accounts for the load in correspondence of the  $\xi$ -th sleeper, and is given by

$$A^\xi(x) = H\left(x - \left(x_\xi - \frac{0.55 \text{ m}}{2}\right)\right) - H\left(x - \left(x_\xi + \frac{0.55 \text{ m}}{2}\right)\right), \quad \xi = 1, \dots, 25 \quad ,$$

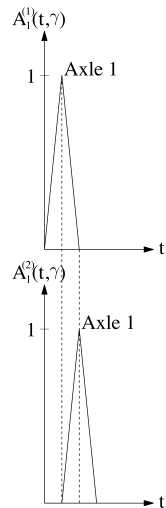
where  $x_\xi$  is the abscissa of the center of gravity of the  $\xi$ -th sleeper, and  $H(\cdot)$  is the Heavyside function.

The time modulating function  $A_\alpha^\xi(t, \gamma)$  allows instead to modulate the pressure value as a function of time and axle speed according to

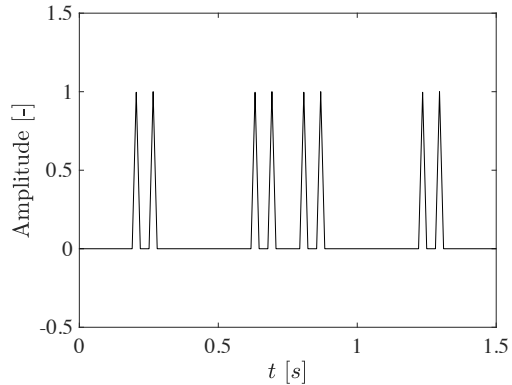
$$A_\alpha^\xi(t, \gamma) = \left[ H\left(t - \frac{x_{\xi-1} + x_\alpha^0}{\gamma}\right) - H\left(t - \frac{x_{\xi+1} + x_\alpha^0}{\gamma}\right) \right] \cdot \left( 1 - \frac{\left| t - \frac{x_\xi + x_\alpha^0}{\gamma} \right|}{\frac{0.65 \text{ m}}{\gamma}} \right), \quad \alpha = 1, \dots, 8, \quad (12)$$

where  $x_\alpha^0$  is the position of the  $\alpha$ -th axle at time  $t = 0$ . Fig. 21 shows that the maximum value of the pressure on a sleeper is attained when the axle is crossing its axis, and it becomes

null when the axle is crossing the axis of the previous or next sleeper, with a linear variation in between. In Fig. 22, the time modulation function of the midspan sleeper for a train speed  $\gamma = 160$  km/h is reported, to show the characteristic history of the external loading for this type of structural systems.



**Figure 21:** Railway bridge: time modulation function. Example related to the 1<sup>st</sup> axle on the 1<sup>st</sup> and 2<sup>nd</sup> sleepers.



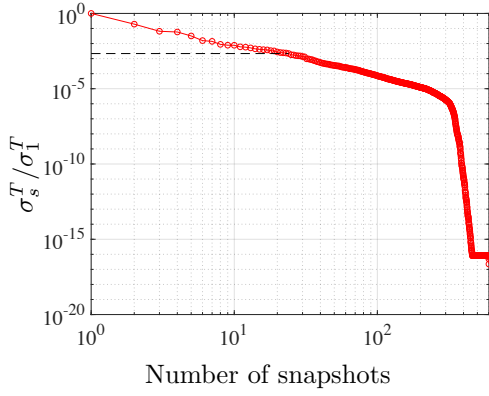
**Figure 22:** Railway bridge: time modulation function. 8 axles passage on the midspan sleeper, at the speed of 160 km/h.

## 4.2 Railway bridge - ROM

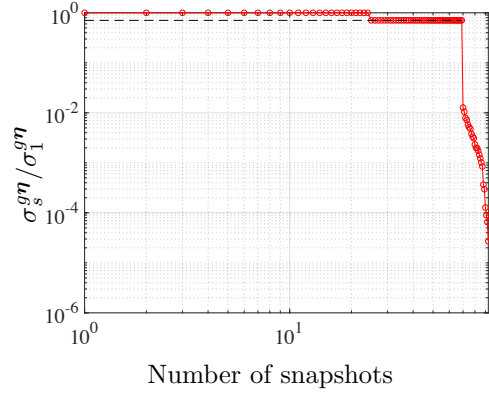
The simulations discussed in what follows have been run on a PC featuring an Intel (R) Core™, i7-2600 CPU @ 3.4 GHz, with a 64 bit operating system and 16 GB RAM. The number of snapshots used to construct the ROM has been set to  $S = 35$ . One could argue that collecting a larger number of snapshots would have further enhanced the representativeness of the ROM, but we have judged the employed number of snapshots as a good trade-off with the high computational cost of each FOM evaluation, as the snapshots collection must be carried out for the entire monitoring time window in order to fully catch the effects of moving loads. Each FOM simulation has required a computing time of about 7 hours, and the ROM construction has required about 10 days.

To attain high approximation capacity, the error tolerance has been set to  $\varepsilon_{tol} = 5 \cdot 10^{-3}$ . In Figs. 23 and 24, the normalized singular values  $\sigma_s^T / \sigma_1^T$  and  $\sigma_s^{g\eta} / \sigma_1^{g\eta}$  are respectively shown for the POD in time and over the parametric space. Confirming what previously stated regarding the need to collect snapshots during the entire monitoring time window, in Fig. 23 the singular values in time are shown to decay only in the final part of the graph, so as the reconstruction error  $\varepsilon$  does.

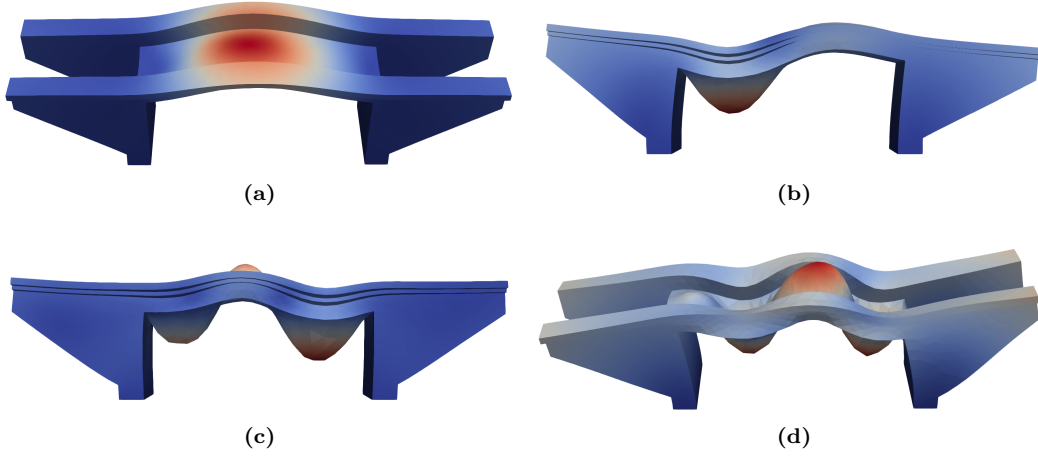
The computational gain obtained with the use of the ROM is even more remarkable than in the previous case study, due to the higher computational complexity of this structure: we have moved from  $M = 15,300$  dofs of the FOM to only  $W = 69$  POD bases. The first 4 POD bases of the bridge are reported in Fig. 25, to show how structural dynamics has been accounted for in the POD-based classification task. These POD bases appear very different if compared with the classical mode shapes; indeed, except for the first one, they do not present symmetries and, due to the peculiarity of the applied load, they are mainly active close to the sleepers area. The computational time required by each simulation has decreased from



**Figure 23:** Railway bridge: POD in time. Descent of the singular values  $\sigma_s^T$  normalised with respect to  $\sigma_1^T$ .



**Figure 24:** Railway bridge: POD over parameters. Descent of the singular values  $\sigma_s^{g\eta}$  normalised with respect to  $\sigma_1^{g\eta}$ .



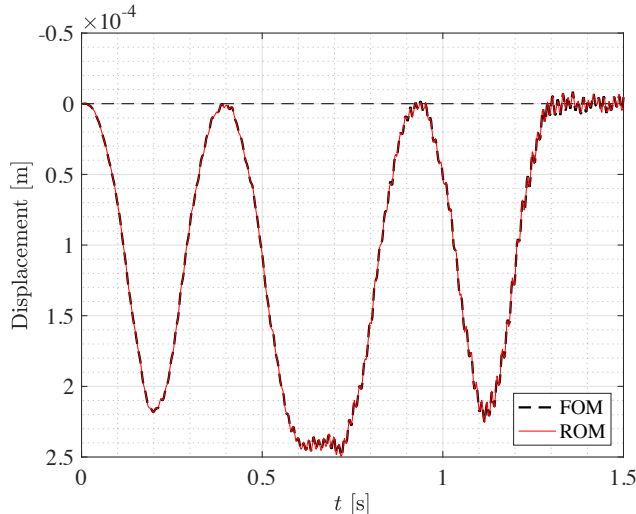
**Figure 25:** Railway bridge: POD bases.

the aforementioned 7 hours for a single FOM solution, to 60 seconds for a ROM solution, with a speed-up of 420. In terms of results, in Fig. 26 the vertical displacement at midspan is reported for  $g = 2$ ,  $\delta = 0.08$ ,  $\gamma = 180$  km/h and  $\beta = 17.325$  ton, so as to assess the ROM accuracy: a noteworthy good approximation capacity is indeed achieved by the ROM, whose response is perfectly superposed to the FOM one. The enrichment in high frequency components of the vertical displacement along the time axis is due to the sequential passage of the axles over the sleepers, and to the absence of damping in the model.

### 4.3 Railway bridge - Classification outcomes

The classifier  $\mathcal{G}$  has been trained and validated on  $I_{tr} + I_{val} = 10,000$  instances, with a ratio 75 : 25, for 1000 epochs. In Figs. 27 and 28 the evolutions of the loss and of the accuracy functions during training are shown: a classification accuracy of 100% has been attained on both the training and validation sets. All the damage scenarios have been therefore perfectly recognized and classified.



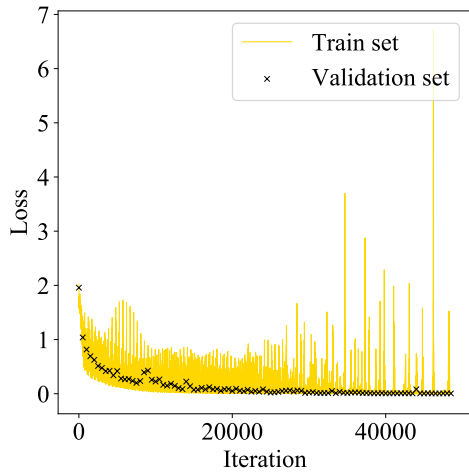


**Figure 26:** Railway bridge: comparison of the vertical displacement time histories at midspan obtained through the FOM and the ROM ( $g = 2$ ,  $\delta = 0.08$ ,  $\gamma = 180$  km/h,  $\beta = 17.325$  ton).

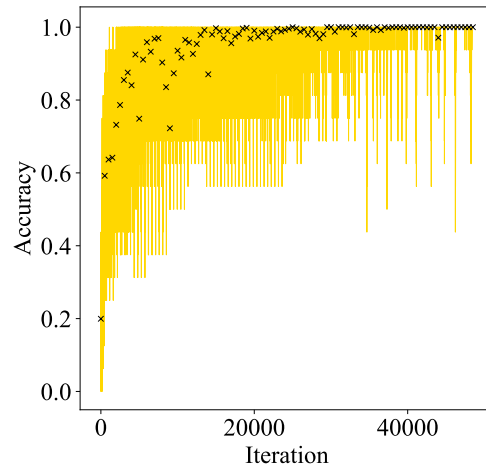
To show the paramount importance of an appropriate deployment of the sensors to measure the structural response to the external loading, a further analysis has been run by neglecting the horizontal recordings  $u_5$  and  $u_6$  in Fig. 20, hence with only information relevant to the  $u_1 - u_4$  time series processed by the classifier. The relevant loss and accuracy evolutions shown in Figs. 29 and 30 highlight that  $\mathcal{G}$  is not able now to recognize all the processed instances correctly. The presence of a systematic classification error is testified also by the confusion matrix in Fig. 31, in which the generalization capabilities of  $\mathcal{G}$  are evaluated against a test set consisting of 42 pseudo-experimental instances simulated with the FOM: a global accuracy of only 88.10% has been obtained. It has emerged that the undamaged scenario is prone to be misclassified as a structural state featuring  $g = 6$ . Such misclassification between scenarios  $g = 0$  and  $g = 6$  is thus due to the missed horizontal dofs in the monitoring system. To further prove this claim, additional tests have been performed on a reduced dataset, characterized by removing scenarios  $g = 5, 6$ ; results in Fig. 32 show how the performance of the classifier returns back to feature a 100% accuracy. The reported overall performances attained by  $\mathcal{G}$  are considered good, especially in view of the high complexity of this example resembling a real monitoring problem.

## 5 Conclusion

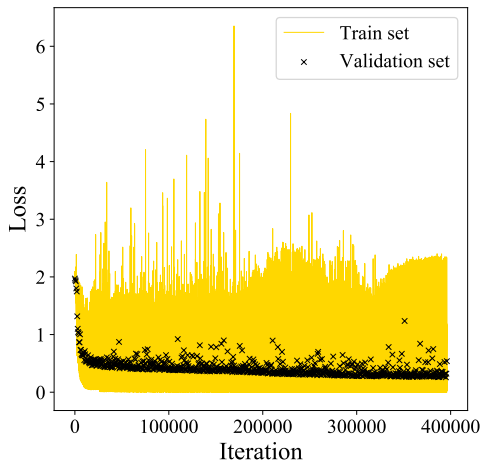
In this work, we have proposed a neural network-based classifier, featuring a fully convolutional network architecture, to move towards online damage localization within a smart structural health monitoring framework. The classifier processes the vibration measurements, recorded by a sensor network deployed over the structure, to identify the current structural state. To overcome the lack of experimental data for civil applications, we have exploited physics-based numerical modeling in order to build offline a large training set of structural responses, accounting for relevant damage scenarios and operational conditions. A parametric model order reduction technique has been next adopted to replace high fidelity, time consuming finite element simulations and speedup the dataset generation. The classifier leverages on the convolutional layers capabilities to automatically extract useful, damage sensitive features from raw data and learn the functional link between such features



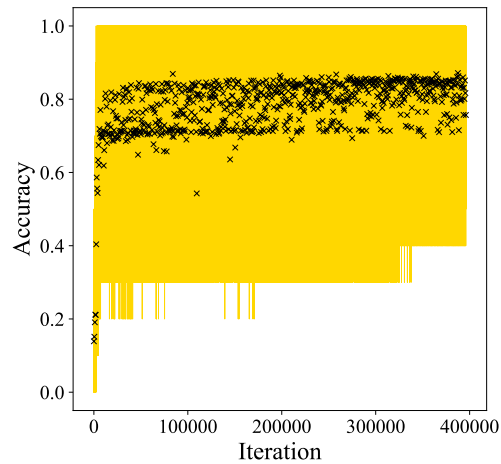
**Figure 27:** Railway bridge: FCN training. Loss function evolution on the training and validation sets.



**Figure 28:** Railway bridge: FCN training. Global accuracy evolution on the training and validation sets.



**Figure 29:** Railway bridge, four sensor monitoring system: FCN training. Loss function evolution on the training and validation sets.



**Figure 30:** Railway bridge, four sensor monitoring system: FCN training. Global accuracy evolution on the training and validation sets.

and the corresponding structural states. The obtained results have confirmed the high potential of the simulation based classification approach to structural health monitoring and of the combined use of parametric model order reduction techniques and deep learning.

In both of the proposed case studies, the global accuracy of classification never falls below 85% , regardless of whether acceleration or displacement measurements are handled. Indeed, the method has proven to be extremely robust in exploring a large parametric dependency and even in recognizing damaged scenarios significantly different from those observed during the training phase.

**Accuracy: 88.10%**

g=0	33.3% 2	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0
g=1	0.0% 0	100.0% 6	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0
g=2	0.0% 0	0.0% 0	100.0% 6	0.0% 0	0.0% 0	0.0% 0	0.0% 0
g=3	0.0% 0	0.0% 0	0.0% 0	100.0% 6	0.0% 0	0.0% 0	0.0% 0
g=4	0.0% 0	0.0% 0	0.0% 0	0.0% 0	100.0% 6	0.0% 0	0.0% 0
g=5	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	100.0% 6	16.7% 1
g=6	66.7% 4	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	83.3% 5
	g=0	g=1	g=2	g=3	g=4	g=5	g=6

Target Class

**Figure 31:** Railway bridge, four sensor monitoring system: FCN testing. Confusion matrix.

**Accuracy: 100.00%**

g=0	100.0% 6	0.0% 0	0.0% 0	0.0% 0	0.0% 0
g=1	0.0% 0	100.0% 6	0.0% 0	0.0% 0	0.0% 0
g=2	0.0% 0	0.0% 0	100.0% 6	0.0% 0	0.0% 0
g=3	0.0% 0	0.0% 0	0.0% 0	100.0% 6	0.0% 0
g=4	0.0% 0	0.0% 0	0.0% 0	0.0% 0	100.0% 6
	g=0	g=1	g=2	g=3	g=4

Target Class

**Figure 32:** Railway bridge, four sensor monitoring system: FCN testing. Confusion matrix with damage scenarios involving only the deck.

Tests have been carried out by adding to the response of the considered digital twins a white noise corruption of varying amplitude, which has been assumed representative of micro-electro mechanical system accelerometer self-noise, and by exploiting different reduced order models of increasing fidelity to build the training datasets. The classification outcomes have shown a slightly decreased global accuracy, featuring a minimum of 78% in the presence of a highly noisy signal (SNR= 20), testifying that the procedure is also rather noise tolerant. Results obtained by exploiting reduced order models generated with a varying value of the error tolerance, have also provided a scheme to assess the effect of the reduced order modeling technique on the classifier performances.

In future works, varying environmental conditions and different excitation sources, such as wind action and low intensity seismicity will be allowed for, by further enlarging the parametric space exploited in the dataset construction. To handle the resulting nonaffine dependency of the numerical arrays on the parametric space, hyper-reduction techniques are going to be exploited. To cope with the need of an optimal sensor placement, we aim to introduce a sensor placement approach to maximize the information effectiveness for the classification task. Further examples are currently under study, in order to validate the offered methodology against suitable experimental settings.

## References

- [1] G. C. Y. Peng, M. Alber, A. Buganza Tepole, W. R. Cannon, S. De, S. Dura-Bernal, K. Garikipati, G. Karniadakis, W. W. Lytton, P. Perdikaris, L. Petzold, and E. Kuhl, "Multiscale modeling meets machine learning: What can we learn?," *Archives of Computational Methods in Engineering*, 2020. <https://doi.org/10.1007/s11831-020-09405-5>.
- [2] F. Chinesta, E. Cueto, E. Abisset Chavanne, J. L. Duval, and F. E. Khaldi, "Virtual, digital and hybrid twins: A new paradigm in data-based engineering and engineered data," *Archives of Computational Methods in Engineering*, vol. 27, pp. 105–134, 2020.
- [3] C. Farrar and K. Worden, *Structural Health Monitoring A Machine Learning Perspective*. Hoboken, NJ: Wiley, 2013.

- [4] C. R. Farrar and K. Worden, “An introduction to structural health monitoring,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1851, pp. 303–315, 2007.
- [5] S. Eftekhar Azam, S. Mariani, and N. Attari, “Online damage detection via a synergy of proper orthogonal decomposition and recursive bayesian filters,” *Nonlinear Dynamics*, vol. 89, no. 2, pp. 1489–1511, 2017.
- [6] S. Eftekhar Azam and S. Mariani, “Online damage detection in structural systems via dynamic inverse analysis: A recursive bayesian approach,” *Engineering Structures*, vol. 159, pp. 28–45, 2018.
- [7] S. Eftekhar Azam, E. Chatzi, and C. Papadimitriou, “A dual kalman filter approach for state estimation via output-only acceleration measurements,” *Mechanical Systems and Signal Processing*, vol. 60-61, pp. 866 – 886, 2015.
- [8] R. Ferrari, D. Froio, E. Rizzi, C. Gentile, and E. N. Chatzi, “Model updating of a historic concrete bridge by sensitivity- and global optimization-based latin hypercube sampling,” *Engineering Structures*, vol. 179, pp. 139 – 160, 2019.
- [9] D. Tibaduiza Burgos, R. Gomez Vargas, C. Pedraza, D. Agis, and F. Pozo, “Damage identification in structural health monitoring: A brief review from its implementation to the use of data-driven applications,” *Sensors*, vol. 20, 733, 2020.
- [10] H. Sohn, K. Worden, and C. R. Farrar, “Statistical damage classification under changing environmental and operational conditions,” *Journal of Intelligent Material Systems and Structures*, vol. 13, no. 9, pp. 561–574, 2002.
- [11] A. Entezami, H. Shariatmadar, and S. Mariani, “Structural health monitoring for condition assessment using efficient supervised learning techniques,” *Proceedings*, vol. 42(1), 17, 2020.
- [12] L. Bull, K. Worden, R. Fuentes, G. Manson, E. Cross, and N. Dervilis, “Outlier ensembles: A robust method for damage detection and unsupervised feature extraction from high-dimensional data,” *Journal of Sound and Vibration*, vol. 453, pp. 126 – 150, 2019.
- [13] A. Entezami, H. Shariatmadar, and S. Mariani, “Fast unsupervised learning methods for structural health monitoring with large vibration data from dense sensor networks,” *Structural Health Monitoring*, vol. 19, no. 6, pp. 1685–1710, 2020.
- [14] A. Entezami, H. Sarmadi, B. Behkamal, and S. Mariani, “Big data analytics and structural health monitoring: A statistical pattern recognition-based approach,” *Sensors*, vol. 20(8), 2328, 2020.
- [15] A. Entezami, H. Shariatmadar, and S. Mariani, “Early damage assessment in large-scale structures by innovative statistical pattern recognition methods based on time series modeling and novelty detection,” *Advances in Engineering Software*, vol. 150, p. 102923, 2020.
- [16] M. H. Rafiei and H. Adeli, “A novel unsupervised deep learning model for global and local health condition assessment of structures,” *Engineering Structures*, vol. 156, pp. 598 – 607, 2018.
- [17] C. R. Farrar, S. W. Doebling, and D. A. Nix, “Vibration-based structural damage identification,” *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, vol. 359, no. 1778, pp. 131–149, 2001.

- [18] T. Taddei, J. Penn, M. Yano, and A. Patera, “Simulation-based classification; a model-order-reduction approach for structural health monitoring,” *Archives of Computational Methods in Engineering*, vol. 25, no. 1, pp. 23–45, 2018.
- [19] C. Bigoni and J. S. Hesthaven, “Simulation-based anomaly detection and damage localization: an application to structural health monitoring,” *Computer Methods in Applied Mechanics and Engineering*, vol. 363, p. 112896, 2020.
- [20] L. Rosafalco, A. Manzoni, S. Mariani, and A. Corigliano, “Fully convolutional networks for structural health monitoring through multivariate time series classification,” *Advanced Modeling and Simulation in Engineering Sciences*, vol. 7, p. 38, 2020.
- [21] A. Quarteroni, A. Manzoni, and F. Negri, *Reduced basis methods for partial differential equations: an introduction*, vol. 92 of *Unitext*. Springer, 2015.
- [22] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, 2015.
- [23] Z. Wang, W. Yan, and T. Oates, “Time series classification from scratch with deep neural networks: A strong baseline,” in *Proceedings of the International Joint Conference on Neural Networks (IJCNN), 14-19 May, (Anchorage, AK)*, pp. 1578–1585, 2017.
- [24] T. Zhang, S. Biswal, and Y. Wang, “Shmnet: Condition assessment of bolted connection with beyond human-level performance,” *Structural Health Monitoring*, vol. 19, no. 4, pp. 1188–1201, 2020.
- [25] L. Rosafalco, A. Corigliano, A. Manzoni, and S. Mariani, “Combined model order reduction and artificial neural network for data assimilation and damage detection in structures,” in *Computational Sciences and Artificial Intelligence in Industry (CSAI), June 12-14, (Jyväskylä, Finland)*, 2019. To appear.
- [26] E. Samaniego, C. Anitescu, S. Goswami, V. Nguyen-Thanh, H. Guo, K. Hamdia, X. Zhuang, and T. Rabczuk, “An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications,” *Computer Methods in Applied Mechanics and Engineering*, vol. 362, p. 112790, 2020.
- [27] G. Capellari, E. Chatzi, and S. Mariani, “An optimal sensor placement method for SHM based on Bayesian experimental design and polynomial chaos expansion,” in *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS), June 5-10, (Athens, Greece)*, pp. 6272–6282, 2016.
- [28] G. Capellari, E. Chatzi, and S. Mariani, “Cost–benefit optimization of structural health monitoring sensor networks,” *Sensors*, vol. 18, no. 7, 2174, 2018.
- [29] C. Bigoni, Z. Zhang, and J. S. Hesthaven, “Systematic sensor placement for structural anomaly detection in the absence of damaged states,” *Computer Methods in Applied Mechanics and Engineering*, vol. 371, p. 113315, 2020.
- [30] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.
- [31] A. L. Samuel, “Some studies in machine learning using the game of checkers,” *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959.
- [32] B. Sudret, G. Defaux, and M. Pendola, “Stochastic evaluation of the damage length in rc beams submitted to corrosion of reinforcing steel,” *Civil Engineering and Environmental Systems*, vol. 24, no. 2, pp. 165–178, 2007.

- [33] Z. Guédé, B. Sudret, and M. Lemaire, “Life-time reliability based assessment of structures submitted to thermal fatigue,” *International Journal of Fatigue*, vol. 29, no. 7, pp. 1359 – 1373, 2007.
- [34] M. D. McKay, R. J. Beckman, and W. J. Conover, “Comparison of three methods for selecting values of input variables in the analysis of output from a computer code,” *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.
- [35] I. Sobol’, “On the distribution of points in a cube and the approximate evaluation of integrals,” *USSR Computational Mathematics and Mathematical Physics*, vol. 7, no. 4, pp. 86 – 112, 1967.
- [36] J. Hesthaven and S. Ubbiali, “Non-intrusive reduced order modeling of nonlinear problems using neural networks,” *Journal of Computational Physics*, vol. 363, pp. 55–78, 2018.
- [37] A. Corigliano and S. Mariani, “Parameter identification in explicit structural dynamics: Performance of the extended Kalman filter,” *Computer Methods in Applied Mechanics and Engineering*, vol. 193, no. 36-38, pp. 3807–3835, 2004.
- [38] A. Teughels, J. Maeck, and G. De Roeck, “Damage assessment by fe model updating using damage functions,” *Computers & Structures*, vol. 80, no. 25, pp. 1869 – 1879, 2002.
- [39] H. M. Hilber, T. J. R. Hughes, and R. L. Taylor, “Improved numerical dissipation for time integration algorithms in structural dynamics,” *Earthquake Engineering & Structural Dynamics*, vol. 5, no. 3, pp. 283–292, 1977.
- [40] L. Rosafalco, A. Corigliano, A. Manzoni, and S. Mariani, “A hybrid structural health monitoring approach based on reduced-order modelling and deep learning,” *Proceedings*, vol. 42(1), 67, 2020.
- [41] G. Kerschen and J. Golinval, “Physical interpretation of the proper orthogonal modes using the singular value decomposition,” *Journal of Sound and Vibration*, vol. 249, no. 5, pp. 849 – 865, 2002.
- [42] S. Haykin, *Neural Networks and Learning Machines*. Upper Saddle River, NJ: Prentice Hall, 2009.
- [43] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *International Conference on Learning Representations (ICLR), 7-9 May*, (San Diego, CA), pp. 1–13, 2015.
- [44] W. Rawat and Z. Wang, “Deep convolutional neural networks for image classification: A comprehensive review,” *Neural Computation*, vol. 29, no. 9, pp. 2352–2449, 2017. PMID: 28599112.
- [45] F. Karim, S. Majumdar, H. Darabi, and S. Harford, “Multivariate LSTM-FCNs for time series classification,” *Neural Networks*, vol. 116, pp. 237–245, 2019.
- [46] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “Wavenet: A generative model for raw audio.” arXiv:1609.03499[cs.SD], 2016.
- [47] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *The IEEE Conference on Computer Vision and Pattern Recognition CVPR, 26 June - 1 July*, (Boston, MA), pp. 1–9, 2015.

- [48] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 18 June - 23 June*, (Salt Lake City, UT), pp. 7132–7141, 2018.
- [49] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proceedings of the 32nd International Conference on Machine Learning, ICML, Lille, France, 6-11 July 2015*, vol. 37 of *JMLR Workshop and Conference Proceedings*, pp. 448–456, JMLR.org, 2015.
- [50] Safran, *VS1000: Single-axis analog accelerometer*, 2016. <https://www.colibrys.com/product/vs1000-vibration-sensor/30s-vs1000-c-04-16-2/>.
- [51] STMicroelectronics, *LIS331DLH: MEMS digital output motion sensor ultra-power high performance 3-axes "nano" accelerometer*, 2009. Rev.3, <https://www.st.com/en/mems-and-sensors/lis3dh.html>.
- [52] A. D’Alessandro, G. Vitale, S. Scudero, R. D’Anna, A. Costanza, A. Fagiolini, and L. Greco, “Characterization of mems accelerometer self-noise by means of psd and allan variance analysis,” in *7th IEEE International Workshop on Advances in Sensors and Interfaces IWASI, 15-17 June, Vieste, Italy*, pp. 159–164, 2017.
- [53] J. R. Evans, R. M. Allen, A. I. Chung, E. S. Cochran, R. Guy, M. Hellweg, and J. F. Lawrence, “Performance of several low-cost accelerometers,” *Seismological Research Letters*, vol. 85, pp. 147–158, 01 2014.
- [54] T. Arvidsson and J. Li, *Dynamic analysis of a portal frame railway bridge using frequency dependent soil structure interaction*. Master thesis, KTH Royal Institute of Technology, 2011.
- [55] M. Ülker-Kaustell, *Some aspects of the dynamic soil-structure interaction of a portal frame railway bridge*. PhD thesis, KTH Royal Institute of Technology, 2009.
- [56] C. Farrar and D. Jauregui, “Damage detection algorithms applied to experimental modal data from the i-40 bridge,” 1996.
- [57] A. Pandey and M. Biswas, “Damage detection in structures using changes in flexibility,” *Journal of Sound and Vibration*, vol. 169, no. 1, pp. 3–17, 1994.
- [58] European Committee for Standardization, “Part 1-1: General rules and rules for buildings,” in *EN 1992-1-1 Eurocode 2: Design of concrete structures*, pp. 57–60, 2003.
- [59] European Committee for Standardization, “Part 2: Traffic loads on bridges,” in *EN 1991-2 Eurocode 1: Actions on structures*, pp. 66–74, 2003.

## MOX Technical Reports, last issues

Dipartimento di Matematica  
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

- 62/2021** Lupo Pasini, M.; Burcul, M.; Reeve, S.; Eisenbach, M.; Perotto, S.  
*Fast and accurate predictions of total energy for solid solution alloys with graph convolutional neural networks*
- 61/2021** Buchwald, S.; Ciaramella, G.; Salomon, J.; Sugny, D.  
*A greedy reconstruction algorithm for the identification of spin distribution*
- 60/2021** Rosafalco, L.; Manzoni, A.; Mariani, S.; Corigliano, A.  
*Fully convolutional networks for structural health monitoring through multivariate time series classification*
- 59/2021** Stella, S.; Regazzoni, F.; Vergara, C.; Dede', L.; Quarteroni, A.  
*A fast cardiac electromechanics model coupling the Eikonal and the nonlinear mechanics equations*
- 58/2021** Tassi, T., Zingaro, A., Dede', L.  
*Enhancing numerical stabilization methods for advection dominated differential problems by Machine Learning algorithms*
- 57/2021** Roberto Piersanti, Francesco Regazzoni, Matteo Salvador, Antonio F. Corno, Luca Dede', Chri  
*3D-0D closed-loop model for the simulation of cardiac biventricular electromechanics*
- 56/2021** Zingaro, A.; Fumagalli, I.; Dede' L.; Fedele M.; Africa P.C.; Corno A.F.; Quarteroni A.  
*A multiscale CFD model of blood flow in the human left heart coupled with a lumped parameter model of the cardiovascular system*
- 54/2021** Ciaramella, G.; Gander, M.J.; Mamooler, P.  
*HOW TO BEST CHOOSE THE OUTER COARSE MESH IN THE DOMAIN DECOMPOSITION METHOD OF BANK AND JIMACK*
- 53/2021** Ciaramella, G.; Mechelli, L.  
*On the effect of boundary conditions on the scalability of Schwarz methods*
- 50/2021** Ciaramella, G.; Vanzan, T.  
*On the asymptotic optimality of spectral coarse spaces*