Chapter 1
On the Detection and Quantification of Nonlinearity via Statistics of the Gradients of a Black-Box Model

Georgios Tsialiamanis and Charles R. Farrar

Abstract Detection and identification of nonlinearity is a task of high importance for structural dynamics. On the one hand, identifying nonlinearity in a structure would allow one to build more accurate models of the structure. On the other hand, detecting nonlinearity in a structure, which has been designed to operate in its linear region, might indicate the existence of damage within the structure. Common damage cases which cause nonlinear behaviour are breathing cracks and points where some material may have reached its plastic region. Therefore, it is important, even for safety reasons, to detect when a structure exhibits nonlinear behaviour. In the current work, a method to detect nonlinearity is proposed, based on the distribution of the gradients of a data-driven model, which is fitted on data acquired from the structure of interest. The data-driven model selected for the current application is a neural network. The selection of such a type of model was done in order to not allow the user to decide how linear or nonlinear the model shall be, but to let the training algorithm of the neural network shape the level of nonlinearity according to the training data. The neural network is trained to predict the accelerations of the structure for a time-instant using as input accelerations of previous time-instants, i.e. one-step-ahead predictions. Afterwards, the gradients of the output of the neural network with respect to its inputs are calculated. Given that the structure is linear, the distribution of the aforementioned gradients should be unimodal and quite peaked, while in the case of a structure with nonlinearities, the distribution of the gradients shall be more spread and, potentially, multimodal. To test the above assumption, data from an experimental structure are considered. The structure is tested under different scenarios, some of which are linear and some of which are nonlinear. More specifically, the nonlinearity is introduced as a column-bumper nonlinearity, aimed at simulating the effects of a breathing crack and at different levels, i.e. different values of the initial gap between the bumper and the column. Following the proposed method, the statistics of the distributions of the gradients for the different scenarios can indeed be used to identify cases where nonlinearity is present. Moreover, via the proposed method one is able to quantify the nonlinearity by observing higher values of standard deviation of the distribution of the gradients for lower values of the initial column-bumper gap, i.e. for “more nonlinear” scenarios.

Keywords Structural health monitoring (SHM) · Structural dynamics · Nonlinear dynamics · Machine learning · Neural networks

1.1 Introduction

In the pursuit of making everyday life safer, humans have extensively tried to model the environment around them. Structures are an important part of the environment, in which humans live. They are man-made and should be safe throughout their lifetime. Structures are exposed to numerous environmental factors, which may cause them to fail. Moreover, during operation, structures are subjected to dynamic loads, which, in time, may cause failure. Such failures will most probably result in economic damage to society and may even result in loss of human lives. Therefore, for the purpose of maintaining structures safe, the field of *structural health monitoring* (SHM) [1] has emerged.

G. Tsialiamanis (✉)
Dynamics Research Group, Department of Mechanical Engineering, University of Sheffield, Sheffield, UK
e-mail: g.tsialiamanis@sheffield.ac.uk

C. R. Farrar
Engineering Institute, MS T-001, Los Alamos National Laboratory, Los Alamos, NM, USA
e-mail: farrar@lanl.gov

M. R. W. Brake et al. (eds.), *Nonlinear Structures & Systems, Volume 1*, Conference Proceedings
of the Society for Experimental Mechanics Series, https://doi.org/10.1007/978-3-031-36999-5_1

The discipline of SHM is the subdiscipline of structural dynamics, which focuses on using data acquired from sensors to evaluate the condition of a structure. There are several tasks that are performed during an SHM project. A convenient categorization of these tasks has been proposed by Rytter in [2], extended in [1] and is given by the hierarchical structure:

1. Is there damage in the system (*existence*)?
2. Where is the damage in the system (*location*)?
3. What kind of damage is present (*type/classification*)?
4. How severe is the damage (*extent/severity*)?
5. How much useful (safe) life remains (*prognosis*)?

The order of the tasks in the above hierarchy may also be viewed as being defined according to their difficulty. The first step, that of identifying whether damage exists in a structure, is considered the most simple one. Several approaches exist, but a quite common one is *outlier detection* [3]. Such approaches could range from fitting simple probability density functions to the data to fitting an *autoencoder neural network* (ANN) to the data to calculate a novelty index [4]. A general strategy for this task is to define a baseline model of the structure, which accurately explains the healthy state, and indicate any state of the structure, which may not be adequately explained by this model, as a potentially damaged state [5].

The second and the third steps of the hierarchy, the localization or the classification of damage, require more specialized tools than the case of detecting damage. As in many cases of modelling systems, there are two types of models to adopt to localize damage, *physics-based* models (e.g. *finite element models* [6]) and *data-driven* or *machine learning* models [7]. The first category refers to models which are built according to one's understanding of the physics of damage, e.g. cracks [8]. The second type of models, machine learning models, is built using mainly data and algorithms which perform pattern recognition to indicate the position of the damage within the structure [9].

The next two steps are considered to be the most difficult ones. To define the extent of damage or how much useful life the structure has, one should have adequate knowledge about the physics of the mechanism of the specific type of damage that affects the structure [10]. In order to perform these processes in a data-driven manner, one should have data available of the evolution of damage of the structure, which might be difficult to acquire. Therefore, a way to deal with problems of the fourth and fifth step of Rytter's hierarchy might be to follow a *population-based structural health monitoring* (PBSHM) framework [11], in order to exploit data from structures in a population, which have already failed, to make predictions about the evolution of damage of existing structures in the same population. In every case, these steps (especially the last one) involve many uncertain parameters, such as the environmental and operational conditions of the structure in the future, making them quite complicated tasks.

All of the steps of the hierarchy have been approached in both a physics-based and a data-driven way and each of these ways has its advantages and disadvantages. On the one hand, physics-based models are based on the underlying physics of the structure and the damage mechanism; therefore, if the formulation of the physics matches or is "close" to the real underlying physics, the models should make accurate predictions about it. However, because of the uncertainties of the environment, capturing accurately the underlying physics for a structure does not ensure that other structures—even identical ones—shall behave the same way.

Data-driven models, on the other hand, do not require definition of analytical expressions regarding the underlying physics of a structure. They rely exclusively on data. An apparent drawback of such models is that it might be challenging to consider them in cases where data from a structure are not available. Moreover, since the main object of study of SHM is damage, data-driven methods are even more difficult to be applied, because, when structures are damaged, they are often quickly repaired, making the access to data from damaged structures even more restricted. Even if one acquires data from the damaged structure, the models trained according to these data may not be efficient for the repaired structure, because the structural parameters may change, as a result of the repair procedure [12].

The current work is mainly focused on the first step of the hierarchy, that of identifying whether a structure is damaged or not. The approach followed herein is motivated by Farrar et al. [13] and Worden et al. [14], where it is assumed that a damaged structure should provide more complex data than a healthy structure—the complexity being measured by several complexity metrics. The approach proposed herein is that of a metric of complexity of the gradients of a neural network model that sufficiently explains the data acquired from a structure. As it will be discussed, the proposed metric is based on the statistics of the gradients of the model and provides a way of discriminating states of the structure that indicate nonlinear behaviour. Moreover, the proposed metric can be calculated for different degrees of freedom of the structure so it may also be used as a tool for the second step of the hierarchy, the localization of damage. Finally, being a scalar metric, in the example presented here, it's magnitude may also be used as an indication of "how intensely nonlinear" a structure is.

The layout of the paper is the following. In the second section, a brief discussion is provided about machine learning in structural dynamics and more specifically SHM and the reasons that it has been widely exploited in many disciplines, including engineering. In the third section, the dataset which is used in the current work is presented and aspects of the specific dataset are discussed. In the fourth section, the proposed methodology is presented and the results of applying it on the dataset. In the final section, conclusions are drawn regarding the proposed algorithm and future steps are discussed.

1.2 Machine Learning for Structural Dynamics

In recent years, machine learning has been widely developed for computer science and many other disciplines as well. One would argue that machine learning gets so much attention because of the impressiveness of the results of its algorithms. For example, machine learning algorithms have achieved generation of real-looking images [15], translation of text into artificially generated images [16] and prediction of the structure of proteins [17], achievements that before the rise of machine learning would have been considered very difficult or even impossible. However, an equally important reason of the success of machine learning is that it has provided an alternative solution to problems that classical approaches have failed.

One of the problems that machine learning has solved quite successfully is that of image recognition [18]. A basic application of image recognition is to simply classify images in classes, regarding the content of the image. The advantage of using machine learning is that the model which takes an image as an input and predicts the class of the image is learnt exclusively from the available data. A classical approach to this problem may have been very difficult to be attained. One may have had to manually identify patterns in the images that point towards each class. Afterwards, one would have to define, also manually, a model that quantifies how plausible it is for every image to belong to a class, according to these patterns. Machine learning and specifically the *convolutional neural network* (CNN) [18] provided a solution to these problems without any need of human intervention in the parameters of the model (only the *hyperparameters* of a machine learning algorithm need manual tuning).

Similarly, for SHM, one of the problems of Rytter's hierarchy can be solved via the use of classification machine learning algorithms. An excellent example of such a solution is presented in [9], where a neural network is built to predict the category of damage on a wing of an aircraft. The damage was simulated by removing panels from the wing and data were acquired during nine different damage cases. The model is trained according to data corresponding to the damaged states and it achieves quite high accuracy in classifying unseen data into the nine damage classes. The image recognition and the damage classification applications reveal a major advantage of machine learning, the bypassing of manually defining a model to perform the desired task. Especially for the damage classification task, one would need extended understanding of the way that the removed panels should affect the recorded data, something that might be quite structure-specific as results in [12] reveal. However, because of the uncertainties of such a phenomenon, machine learning proves a far more convenient and effective solution.

The two examples mentioned in the previous paragraphs refer to problems whose solution in a traditional way may have been infeasible, i.e. the definition of closed form equations that would classify the images into the desired classes or the sensor signals into the damage classes. However, machine learning can provide solutions in a similar manner in case of problems for which one may have available equations. An example of such a problem is that of system identification [19]. Quite often, one may have a parametric set of equations for a system but the set of equations may suffer from *epistemic uncertainty*, i.e. the equations may not resemble the completely true underlying physics of the problem. Therefore, such a set of equations may not be able to sufficiently explain the phenomenon and make accurate predictions. In such cases, a *black-box* modelling approach can be followed. A black-box model is simply a model that learns exclusively from data the relationship between some input and some output quantities of interest. A characteristic example of such a model is a neural network [7].

In the current work, a functionality of neural networks which has become popular in the community of scientific machine learning is exploited. This functionality refers to the calculation of gradients of neural network models with respect to their inputs. Differentiation is a very important part of defining and training a neural network, because *backpropagation*, the algorithm used to train a neural network, is based on calculating the derivatives of a *loss function* with respect to the tunable parameters of the network. Recently, the calculation of derivatives has been extended to the inputs of the neural network because of the definition of *physics-informed neural networks* (PINN) [20]. For such a type of neural networks, the researchers impose their knowledge of the underlying physics of the studied system as part of the loss function. This knowledge often involves relationships of the derivatives of the defined model; therefore, the calculation of derivatives with respect to input quantities has been utilized. In the case of PINNs the gradient is used as a means of imposing physical knowledge into the model. However, in the current work the gradients are studied after training the model as a way of analysing the model, aiming at a better study and higher explainability of the model.

1.3 A Three-Storey Building Dataset

The dataset which is used in the current work comes from a three-storey structure, shown in Fig. 1.1, which was tested in the laboratory. The structure was considered in 17 different states, but the 14 of those are of interest of the current work. State 1 is considered the baseline state, states 2–9 are considered to be undamaged, and states 10–14 are considered the damaged ones. The difference between states 2–9 from the baseline state is regarding the stiffness of different columns of the structure or some added mass on a floor. For example, state 3 has an extra 1.2 kg on the first floor and state 6 has a stiffness reduction on the front column of the first floor. In large scale structures, such stiffness reductions could be observed in cases of temperature differences between two parts of the structure, e.g. a bridge whose deck is heated by the sun, but the underneath structure is heated slower. The damage for states 10–14 is simulated by engaging the bumper and the column between the second and the third floor, shown within the dashed box in Fig. 1.1. A damage type, which could be simulated by such a column-bumper setup, could be a breathing crack. The difference between the damaged states is the width of the initial gap between the column and the bumper, and for higher number of state, the gap becomes smaller, which can be considered an increase in how harsh the nonlinearity of the structure is, because the smaller the gap, the more often the two elements will collide and probably with higher velocity. Namely, the gap was 0.20 mm, 0.15 mm, 0.13 mm, 0.10 mm and 0.05 mm for states 10–14, respectively.

The structure was excited using a white noise signal on its base. The acquired data came from accelerometers placed on each floor and the base. For every state of the structure, the experiment was repeated 50 times, and for every repetition the excitation and the recording of the data lasted 25.6 s and the data were recorded in a sampling frequency of 320 Hz, resulting in 8192 data points for each one of the 50 repetitions. Therefore, for every state, a dataset of 409600 points is available.

The aim of the current work is to define a metric which shall be used to identify the existence of damage and which shall have a higher value for greater extent of damage. As mentioned, in the current dataset, the damaged states are considered the ones which have a nonlinearity. Furthermore, the nonlinearity of the structure is considered to be increasing as the gap between the column and the bumper is decreasing. On the contrary, states 2–9 are different compared to the baseline state 1, but are still linear and not considered as damaged. As a result, the desired metric should be a metric of nonlinearity which should have a higher value as the structure becomes more nonlinear or, in the current case, as the gap between the bumper and the column becomes smaller. As a first step, a model is desired, which can be used to make accurate predictions and be studied in order to define a metric with the aforementioned properties.

Following a physics-based framework, such a model, which would be fitted to the data, would have a predefined form, for example:

$$[M]\{\ddot{\mathbf{y}}\} + [C]\{\dot{\mathbf{y}}\} + [K]\{\mathbf{y}\} = F \tag{1.1}$$

where $[M]$, $[C]$, $[K]$ are the mass, damping and stiffness matrices, respectively, $\{\ddot{\mathbf{y}}\}$, $\{\dot{\mathbf{y}}\}$, $\{\mathbf{y}\}$ are the acceleration, velocity and displacement vectors and F is the forcing vector of the system. A major problem of such an approach is that if the

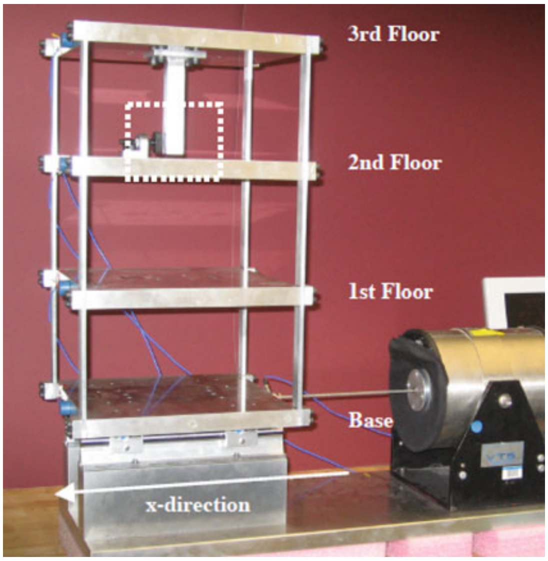


Fig. 1.1 Experimental setup of three floors and the bumper nonlinearity between the second and third floor shown in the dashed box [21]

predefined equation does not match the underlying physics of the structure, the model may not be sufficient to make accurate predictions. Although the structure is considered linear for states 1–9, there might be nonlinearities, for example, in the joints or the damping.

Another approach for a model that can be studied in order define the desired metric is to define a machine learning model. For this purpose, in the current work, a neural network is chosen as such a model. The reasons to use a neural network are two. First, a neural network is a universal approximator [22], making it a great model in the case of unknown underlying physics. An example of a model which does not have such a property is a polynomial regression model of predefined order. The second reason is that a neural network can be recalibrated from its baseline state according to data from a new state. The desired approach herein is to define a model for the baseline state and to recalibrate it for new data. Following this approach, the change of the model can be studied and compared to the baseline model to define the aforementioned metric.

1.4 Statistics of the Model Gradients as a Nonlinearity Metric

For a single-degree-of-freedom system, a general equation that can be considered for dynamic systems is given by:

$$m\ddot{y} + c\dot{y} + ky + g(y, \dot{y}) = F(t) \tag{1.2}$$

where m, c, k are the mass, damping and stiffness of the oscillator, $\ddot{y}$, $\dot{y}$, y are the acceleration, velocity and displacement of the system, $F(t)$ is the force signal applied to the system and $g(y, \dot{y})$ is a function of nonlinear terms of y. A common way to solve such an equation is in a discrete-time framework, by replacing $\ddot{y}$ and $\dot{y}$ by their finite difference approximation, i.e. $\dot{y} = \frac{y_t - y_{t-1}}{dt}$ and $\ddot{y} = \frac{\dot{y}_t - \dot{y}_{t-1}}{dt}$. The solution then is a *one-step-ahead* (OSA) model having the form:

$$y_t = f(y_{t-1}, y_{t-2}, \dots y_{t-l}) \tag{1.3}$$

where f is a model and l is the *lag*, which is number of timesteps before the timestep t that are used as inputs to the model. For the linear case, the equation above becomes $y_t = (2 - \frac{c\Delta t}{m} - \frac{k\Delta t^2}{m})y_{t-1} + (\frac{c\Delta t}{m} - 1)y_{t-2} + \frac{\Delta t^2}{m}F_{t-1}$ [1].

From Eq. (1.3) the gradients of y_t with respect to y_{t-i} can be calculated as:

$$\frac{\partial y_t}{\partial y_{t-i}}|_{y_0} = \frac{\partial f}{\partial y_{t-i}}|_{y_0} \quad i = 1, 2, \dots l \tag{1.4}$$

where y_0 is the value of y_{t-i} for which the derivative is calculated. The distribution of these derivatives is a meaningful object which characterizes the system. For the linear case, the value of $g(y, \dot{y})$ of Eq. (1.2) is zero, making the relationship of Eq. (1.3) linear and the value of the aforementioned gradients constant. The distribution of the gradients should be a Kronecker delta. However, since the model f is a statistical model and noise will be present in the data, the distribution is expected to be a quite narrow Gaussian-like distribution.

For a wide range of types of nonlinearity, the distribution of the gradients is expected to spread. Quite often, nonlinear systems exhibit nonlinear behaviour for larger values of displacements, e.g. the Duffing oscillator. In these cases, the value of $g(y, \dot{y})$ of Eq. (1.2) is not equal to zero, but as the magnitude of y increases, the contribution of g to the movement of the system becomes more evident. For an example of the Duffing oscillator $g(y, \dot{y}) = k_3 y^3$; therefore, the derivative will have constant terms, because of the linear part of the system, and nonlinear terms proportional to y^2, which for small values of y is almost equal to zero, but for larger values affects the value of the derivative. As a result, the distribution of the derivatives will not be as narrow as in the linear case.

Using the available dataset from the three-storey building, an OSA model can be defined using a neural network. The model can be defined as:

$$\{\ddot{\mathbf{y}}_t\} = f(\{\ddot{\mathbf{y}}_{t-1}\}, \{\ddot{\mathbf{y}}_{t-2}\}, \dots \{\ddot{\mathbf{y}}_{t-l}\}) \tag{1.5}$$

where $\{\ddot{\mathbf{y}}_t\} = [y_t^1, y_t^2, y_t^3, y_t^4]$ is the vector of accelerations of the four degrees of freedom for timestep t, f is the neural network model and l is the *lag*, which is number of timesteps before the timestep t that are used as inputs to the model. To emulate a real situation where the forcing vector is not available to use for modelling, it is not used as an input to the model; however, the accelerations of the base can be considered the forcing of the rest of the building, as in an earthquake situation.

Having defined a model such as the one defined in Eq. (1.5) allows studying its gradients with respect to its inputs. More specifically, for the acceleration of the ith degree of freedom, the derivative with respect to the acceleration of the same degree of freedom but of the previous timestep is defined as:

$$\frac{\partial y_t^i}{\partial y_{t-1}^i}|_{y_0^i} = \frac{\partial f^i}{\partial y_{t-1}^i}|_{y_0^i} \tag{1.6}$$

where f^i is the ith output of the neural network model of Eq. (1.5). The quantity of the above equation is calculated for a specific value of $y_{t-1}^i = y_0^i$. Therefore, by calculating the derivative for several values of y_{t-1}^i one can calculate the distribution of these derivatives.

For the three-storey structure, a simple feedforward neural network is chosen as the model, which will be fitted on the available data. The considered neural network had three layers, an input layer, a hidden layer and an output layer. The dimensionality of the input layer is defined by the size of the lag, which is one of the hyperparameters of the algorithm and should be optimized. The size of the output layer can be equal to the number of degrees of freedom of the system, according to Eq. (1.5), but to allow the model to be specialized separately to each degree of freedom, in the current work, different models are trained for the accelerations of each degree of freedom, i.e. $\ddot{y}_t^i = f(\{\ddot{\mathbf{y}}_{t-1}\}, \{\ddot{\mathbf{y}}_{t-2}\}, \ldots \{\ddot{\mathbf{y}}_{t-l}\})$.

The size of the hidden layer is also a hyperparameter. A common approach for the size of the hidden layer is to train the neural network using a part of the available dataset, called the training dataset, and calculate the error on a second part of the dataset, called the validation dataset. Then the optimal size of the hidden layer is considered one with which the minimum error on the validation dataset is achieved. Although this is an effective strategy to optimize the size of the hidden layer, in the current work, it is predefined by the authors. The size is predefined because the model is fitted on the baseline state data and then shall be recalibrated according to the data from the new states. Having the same model offers the chance for a comparison between the distribution of the gradients of the baseline state and that of the new states. Therefore, the size of the hidden layer should be chosen in a way to allow the neural network to have approximation capabilities for a wide range of linear and nonlinear states. The size is chosen to be 100 neurons, which is considered a decent size. In every case, the model is tested on unseen data to ensure that the error on these data is not high, i.e. it is not overfitted to the training data.

As mentioned, proper training requires splitting the complete dataset in a training and a validation dataset. In order to test the algorithm on unseen data, a third part is considered, the testing dataset. The whole procedure followed to train the baseline model was to split the whole dataset of state 1 into the three aforementioned parts. Afterwards different values for the lag were studied, to train the model on the training dataset and to pick the one model that performs best on the validation dataset. It was noted that a lag value of 2 was sufficient for the model to perform satisfactorily on the validation dataset.

The performance of the model was examined using the *normalized mean-square error* (NMSE), given by $\frac{100}{N\sigma_y^2}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$, where N is the number of available samples in the dataset, σ_y is the standard deviation of the target values, $\hat{y}_i$ are the predictions of the model and y_i are the actual target values. The NMSE is a convenient measure of error in regression problems, since it provides an objective measure of the accuracy, regardless of the scale of the data. NMSE values close to 100% indicate that the model does no better than simply using the mean value of the data, while the lower the value the better the model is calibrated. From experience, values of NMSE lower than 5% indicate a well-fitted model, and values lower than 1% show an excellent model. For the applications presented here, the NMSE was for every state lower than 5% for all three datasets (training, validation, testing), indicating that the models were not overfitted.

Another aspect of training, which is considered important in the current work, was the use of *regularization* [23]. Regularization is often used to prevent overfitting by enforcing smoother mappings between the input and the output quantities of the neural network. In the current work, its benefit is considered to be the smoothness of the mapping, which shall allow the gradients to smoothly change and provide more smooth distributions. For the same reason, the activation functions of the neural network are *hyperbolic tangent* (tanh) activation functions for the hidden layer and a linear activation function for the output layer.

After fitting the model to the baseline state, for every new state, the model was recalibrated. This procedure is simply the training of the neural network considering as the initial state of its trainable parameters the trained-on-the-baseline-data model. After retraining for each different state, the gradients were calculated for the accelerations of every degree of freedom with respect to every input variable (the lagged accelerations). For these distributions three moments were calculated: the standard deviation, the skewness and the kurtosis. The standard deviation and the inverse of the kurtosis exhibited good results and are presented herein; the skewness did not exhibit any worth-presenting results. More specifically, the mean values of the moments are used, given by:

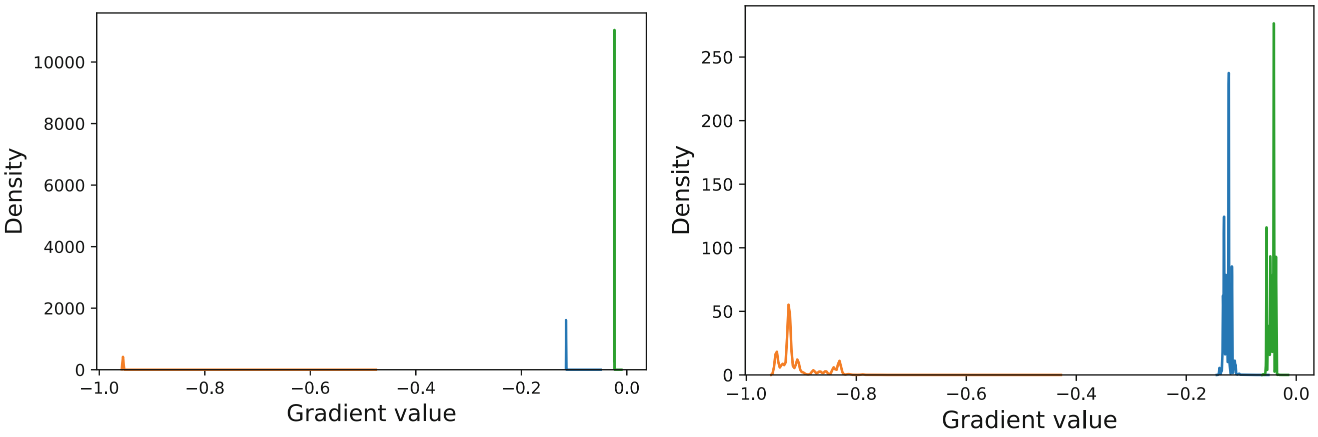


Fig. 1.2 Distribution of the gradients of the model for the predictions of the third floor with respect to the accelerations of the previous timestep of the base (blue line), the first floor (orange line) and the second floor (green line). The distributions are shown for two cases, for the baseline state (left) and for state 14 (right), the most nonlinear case

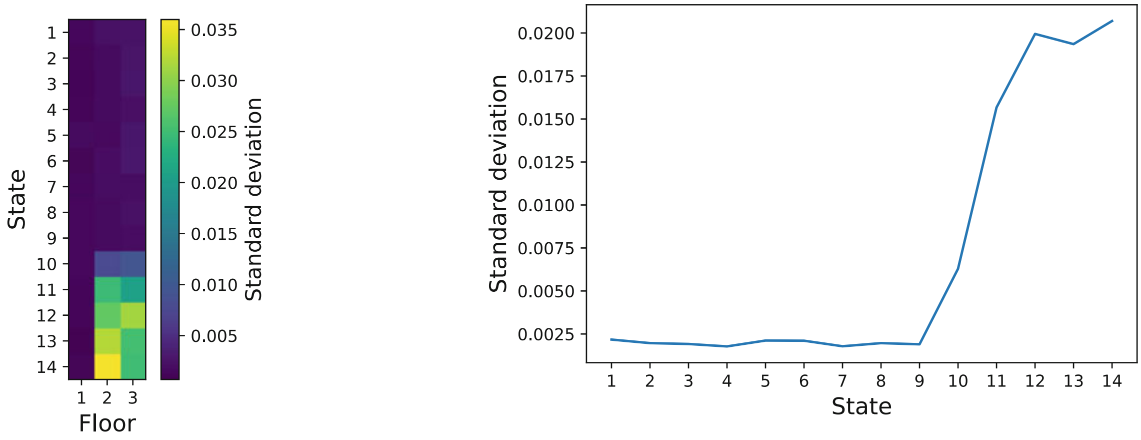


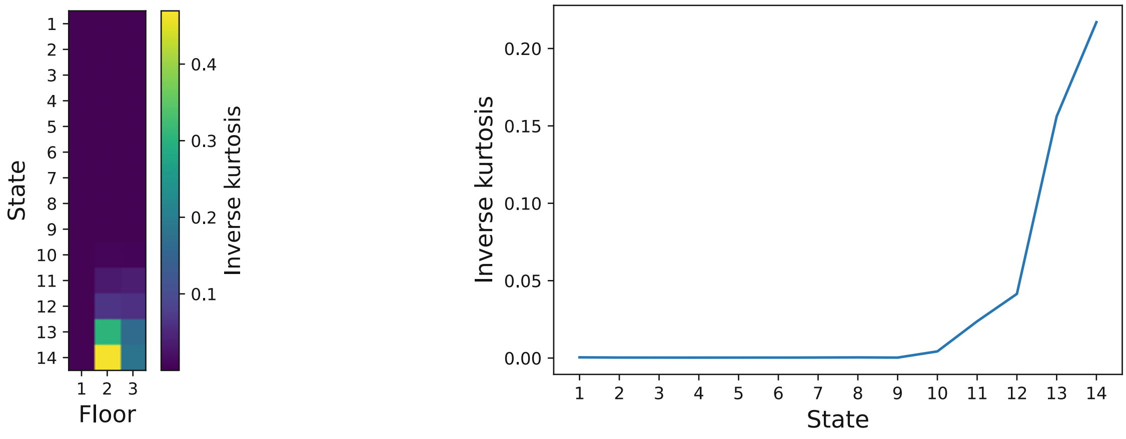
Fig. 1.3 The values of the metric of Eq. (1.7) using the standard deviation as M, for the accelerations of the three floors of the structure (left) and the average value of all three floors of the metric for the different states (right)

$$\bar{M}^i = \frac{1}{D}\sum_{j=1}^{D} M\left[\frac{\partial f^i}{\partial y_{t-1}^j}|_{y_0^j}\right] \quad i = 1, 2, \ldots n_{dof} \tag{1.7}$$

where M is a moment (standard deviation or inverse kurtosis herein), D is the dimensionality of the input, n_{dof} is the number of degrees of freedom of the system and y_0^j are the points for which the gradients are calculated—these points in the current work are all the available points in the three-storey-building dataset. Essentially, the moments are calculated separately for the distributions of the gradients with respect to every input quantity and the average of these values is used as a metric.

At first, the aforementioned distributions are presented to provide a visual confirmation of the intuition presented in the previous paragraphs. Since samples of the gradients are available, a probability density function (PDF) needs to be defined for the purpose of visualizing the distributions. The PDF was calculated using a kernel density estimation and the Silverman's algorithm [24]. The results for the baseline state (state 1) and for the most nonlinear case (state 14) are presented in Fig. 1.2. From the plots, it is clear that for the undamaged state the distribution of the gradients is much more peaked than in the case of the nonlinear structure, where the expected spread of the distributions is clearly observed.

Subsequently, using Eq. (1.7) two metrics are calculated. The first metric is the standard deviation of the distributions of the gradients. The standard deviation is a metric of spreadness of the distribution. Higher values of standard deviation mean a more spread distribution. In Fig. 1.3 on the left, using Eq. (1.7), the average standard deviation of the model of the acceleration of the three floors is shown for different states. On the right of the figure, the mean values of the rows of the left-side plot are shown. The results reveal that using the standard deviation as a metric, the nonlinear states can be identified.



(a)

(b)

Fig. 1.4 The values of the metric of Eq. (1.7) using inverse kurtosis as M, for the accelerations of the three floors of the structure (left) and the average value of all three floors of the metric for the different states (right)

Moreover, a monotonic increase on the value of the metric is observed, as the column-bumper gap becomes smaller, except for state 13, where a small decrease is observed, compared to states 12 and 14. Another interesting aspect of the metric is that its value is increasing mainly for the second and third floors, where the nonlinearity is introduced, making it a candidate metric for localization of damage.

The second metric examined is the inverse of the kurtosis of the distributions. Kurtosis is the fourth moment of the distributions and is a metric of how concentrated the mass of the distribution is around its mean. Higher values of kurtosis mean a greater concentration around the mean value and lower values mean more spread values further away from the mean. Therefore, because of its definition, kurtosis is higher for more peaked distributions, and its inverse is lower, making it a metric whose value increases as the distributions spread. The results for the inverse kurtosis are presented in a similar way to the standard deviation results in Fig. 1.4. The results in this case seem even better than in the case of using the standard deviation. For the undamaged states, the metric is almost equal to zero and it monotonically increases only for the second and the third floors. The average also exhibits better results than the standard deviation, being monotonically higher for smaller values of the column-bumper gap.

1.5 Conclusions

In the current paper, a metric for detecting and quantifying the nonlinearity of a structure is presented. The metric is based on the calculation of the statistics of the gradients of a model, which is trained as a one-step-ahead model for the data acquired from a structure. The model is built for the baseline undamaged state of a structure, having as input lagged accelerations of the structure and as outputs the values of the acceleration one step ahead in time. For new testing states of the structure, the model is recalibrated to the newly acquired data. The gradients for every output of the model are then calculated with respect to the different inputs of the model for the available data samples of the dataset. The distribution of the values of these gradients is then studied. It is expected that for linear cases the distribution shall be quite peaked and as damage evolves within the structure, it shall cause nonlinear effects and affect the distribution of the gradients. The expected effect is a spread of the values. The spread is studied in the current work using two statistical moments, the standard deviation and the inverse of the kurtosis of the distributions.

The aforementioned methodology is tested on a dataset from a three-storey experimental structure. The structure is tested at 14 different states, 9 of which are considered undamaged and the rest 5 are considered damaged. The difference between the undamaged states is in the stiffness of the columns of the structure, and, for the damaged states, the damage is simulated as an added column-bumper setup between two floors of the structure, with varying initial gap between the two elements.

The proposed methodology was applied on the data from the experimental structure and the results revealed that the proposed metrics works as intended. The value of the metric is higher for the damaged cases, making it a tool for identification of damaged states. As the gap between the column and the bumper becomes smaller, making the structure more intensely nonlinear, the metric also increases, which means that the metric could potentially be used for the definition of the severity of damage. Furthermore, the metric is higher for the distributions of the gradients of the accelerations of the two floors, between which the column and the bumper are placed, making it also a potential metric for damage localization.

Further validation of the methodology is needed. However, being tested on experimental data the methodology proves quite efficient. Real-life structures which are designed to operate mainly in the linear region of their members exist, e.g. nuclear plants. Therefore, a similar situation with the experimental setup of the current work could be encountered in such structures. In these cases a methodology similar to the one presented here could be used for identification, localization and quantification of damage within the structure. Moreover, in future work, analysis of the form of the distributions could be made to infer the type of nonlinearity and other comparisons between the different-state distributions could be made to extract further information about the source of the nonlinearity or even to differentiate the linear cases.

Acknowledgments G. Tsialiamanis would like to acknowledge the support of the UK Engineering & Physical Sciences Research Council (EPSRC) via the Programme Grant EP/R006768/1 and European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 764547. For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) licence to any author accepted manuscript version arising.

References

1. Farrar, C.R., Worden, K.: Structural Health Monitoring: A Machine Learning Perspective. Wiley, New York (2011)
2. Rytter, A.: Vibrational Based Inspection of Civil Engineering Structures. PhD thesis, Aalborg University, Denmark (1993)
3. Barnett, V., Lewis, T.: Outliers in Statistical Data. Wiley, Chichester (1994)
4. Worden, K., Manson, G., Allman, D.: Experimental validation of a structural health monitoring methodology: Part I. Novelty detection on a laboratory structure. J. Sound Vib. **259**(2), 323–343 (2003)
5. Sohn, H., Farrar, C.R.: Damage diagnosis using time series analysis of vibration signals. Smart Mater. Struct. **10**(3), 446 (2001)
6. Bathe, K.-J.: Finite Element Procedures. Klaus-Jurgen Bathe, Berlin (2006)
7. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Berlin (2006)
8. Agathos, K., Chatzi, E., Bordas, S.: Multiple crack detection in 3d using a stable XFEM and global optimization. Comput. Mech. **62**(4), 835–852 (2018)
9. Manson, G., Worden, K., Allman, D.: Experimental validation of a structural health monitoring methodology: Part III. Damage location on an aircraft wing. J. Sound Vib. **259**(2), 365–385 (2003)
10. Corbetta, M., Sbarufatti, C., Giglio, M., Todd, M.D.: Optimization of nonlinear, non-Gaussian Bayesian filtering for diagnosis and prognosis of monotonic degradation processes. Mech. Syst. Signal Process. **104**, 305–322 (2018)
11. Gardner, P., Bull, L.A., Gosliga, J., Dervilis, N., Cross, E.J., Papatheou, E., Worden, K.: Population-Based Structural Health Monitoring. In: Structural Health Monitoring Based on Data Science Techniques, pp. 413–435. Springer, New York (2022)
12. Gardner, P.A., Bull, L.A., Dervilis, N., Worden, K.: Challenges for SHM from structural repairs: An outlier-informed domain adaptation approach. In: Data Science in Engineering, vol. 9, pp. 75–86. Springer, New York (2022)
13. Farrar, C., Park, G., Worden, K.: Complexity: A new axiom for structural health monitoring? Technical report, Los Alamos National Laboratory, New Mexico (2010)
14. Worden, K., Dervilis, N., Farrar, C.: Applying the concept of complexity to structural health monitoring. Struct. Health Monit. **2019** (2019)
15. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S. Courville, A., Bengio, Y.: Generative adversarial nets. Adv. Neural Inf. Proces. Syst. **27** (2014)
16. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 (2022)
17. Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al.: Highly accurate protein structure prediction with alphafold. Nature **596**(7873), 583–589 (2021)
18. LeCun, Y., Bengio, Y., et al.: Convolutional networks for images, speech, and time series. The Handbook of Brain Theory and Neural Networks **3361**(10), 1995 (1995)
19. Rogers, T.J., Worden, K., Cross, E.J.: On the application of gaussian process latent force models for joint input-state-parameter estimation: With a view to Bayesian operational identification. Mech. Syst. Signal Process. **140**, 106580 (2020)
20. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. **378**, 686–707 (2019)
21. Figueiredo, E., Park Gyuhae, G., Figueiras, J., Farrar, C., Worden Keith, K.: Structural health monitoring algorithm comparisons using standard data sets. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States) (2009)
22. Csáji, B.C., et al.: Approximation with artificial neural networks. Faculty of Sciences, Eotovs Lorand University, Hungary **24**(48), 7 (2001)
23. Krogh, A., Hertz, J.: A simple weight decay can improve generalization. Adv. Neural Inf. Proces. Syst. **4** (1991)
24. Silverman, B.W.: Using kernel density estimates to investigate multimodality. J. R. Stat. Soc. Ser. B Methodol. **43**(1), 97–99 (1981)