Received: 1 September 2021 | Revised: 8 February 2022 | Accepted: 13 March 2022

DOI: 10.1002/rnc.6124

RESEARCH ARTICLE

WILEY

Learning methods for structural damage detection via entropy-based sensors selection

Francesco Smarra | **Jimmy Tjen** | **Alessandro D'Innocenzo**

Department of Information Engineering, Computer Science and Mathematics, Università degli Studi dell'Aquila, L'Aquila, Italy

Correspondence
Francesco Smarra, Department of Information Engineering, Computer Science and Mathematics, Università degli Studi dell'Aquila, Via Vetoio, Coppito, 67100 L'Aquila, Italy.
Email: francesco.smarra@univaq.it

Funding information
European Commission, Grant/Award Numbers: H2020-ECSEL- 2019-1-IA, H2020-ECSEL-2019-2-RIA; Italian Government, Grant/Award Number: Cipe resolution n.135

Abstract

In this article the problem of data-driven structural damage detection is considered exploiting historical data collected from a structure. First, a novel technique based on Kalman filtering and on a combination of regression trees theory from machine learning and auto-regressive system identification from control theory is derived to build switching models that can be used to detect structural damages. A technique is also proposed leveraging principal component analysis together with the poly-exponential approach to create nonlinear models to be used for structural damage detection. Finally, a novel sensors selection algorithm based on the notions of entropy and information gain from information theory is developed to reduce the number of sensors without affecting or even improving, as it happens in our experimental setup, the model accuracy. The presented techniques are validated on three independent experimental datasets, showing that the proposed algorithms outperform previous and classical approaches, improving the prediction accuracy and the damage detection sensitivity while reducing the number of sensors.

KEYWORDS

fault detection, machine learning, regression trees, sensor selection, structural damage detection, structural health monitoring

1 | INTRODUCTION

Structural health monitoring systems are systems that involve the use of sensors placed on a structure together with algorithms to continuously monitor the structure health. In this context, enabled by recent advances in sensors technology, as well as telecommunication infrastructures, civil engineering structures are expected to be equipped with large networks of sensors (e.g., accelerometers), collecting nominal and non-nominal (i.e., when a damage on the structure occurs) data related to the structural health.[1,2] Such collected data can be used to derive, using system identification techniques, a mathematical model of the nominal (and possibly faulty) structure behavior, so that existing techniques for early damage detection can be applied, see for example, References 3-5.

Motivation and related work for SHM systems modeling and damage detection. This data-driven strategy is particularly relevant when the behavior of a structure is too complex to be derived using physics-based modeling or finite element methods. In this case, the cost/effort to derive a mathematical model is in most cases impractical or even impossible,

hence data-driven approaches are necessary.[6] In the last years, data-driven approaches for SHM systems have been widely investigated, however lot of work still has to be done.[7] In particular, a first challenge that arises is about the modeling of the dynamics of the structures under investigation and the consequent damage detection methodologies involved.

A widely explored area in this sense consists on the use of deep learning methodologies: the authors in References 8-11 proposed data-driven approaches based on neural networks (NNs) algorithms in different scenarios to detect faults occurrence in structures. In general, they have demonstrated that the proposed NNs-based algorithms are efficient in detecting faults existence on structures. However, a main well known drawback of such approaches is that NNs provide quite complex models. Although new technologies are helping the widespread of NNs by overcoming the computational complexity problems,[6] in some cases simpler models can be needed. Furthermore, in terms of application of standard fault detection techniques, simpler or even linear models can be better depending on the application under consideration.

In this respect, much simpler and well established techniques that can address such an important issue, and that are widely used in the SHM modeling context, are the partial least squares (PLS)[12] and the principal component analysis (PCA).[13] PLS is a standard technique that derives a set of linear combinations of the inputs to produce a regression model for the output estimation. The widespread adoption of the PLS in different fields of application, for example, in the SHM one,[14] is motivated by its simplicity. However, an obvious drawback of such technique is due to its capability to well approximate the dynamics of complex systems that can also exhibit a nonlinear behavior: we will indeed show later on that the damage detection sensitivity is worse than in the other methods. PCA consists of an orthogonal projection of a dataset to a lower dimensional space such that the variance is maximized. Through the dimension reduction PCA can unveil a much simpler pattern that can be extracted using eigen value decomposition (EVD) via the sample covariance matrix.[15] This pattern is in general very sensitive to structural changes (e.g., damages), and can be used for structural damage detection. There are several research papers based on this paradigm, see for example, References 16-19. More precisely, Krishnan et al.[16] performed damage detection by on-line estimating with a Kalman filter the time evolution of the parameters of a PCA-based autoregressive (AR) model of the structure; Lakshmi et al.[17] focused on the output response generated by the PCA-based AR model; Dunia et al.[18] suggested the use of PCA for sensor fault identification by reconstructing each variable using iterative substitution and optimization; Kesavan et al.[20] proposed the idea of damage detection by combining PCA analysis and wavelet transformation; Nguyen et al.[19] addressed the problem on nonlinearity in PCA using kernel-based PCA. In general, all these techniques demonstrate the capability of PCA-based models to detect damages of a structure. However, while the PCA is a versatile and robust method, it also has some limitations. First, PCA is performed by calculating the EVD of the data: hence, the complexity of the algorithm increases as the number of sensors increases. In particular, the algorithm complexity of eigenvalue computation for an $n \times n$ matrix is $O(n^3)$ with standard SVD algorithms.[21] As a consequence, even PCA can be computationally intractable or inefficient when the number of sensors is very large. Another limitation is that the principal component (PC) of PCA is given as a linear combination of the original data: hence, if the linearity assumption is not satisfied PCA does not provide the desired prediction accuracy.[22] Furthermore, on high multidimensional data, upward bias in sample eigenvalue and inconsistencies of sample eigenvectors can appear, and PCA provides unexpected results.[23]

Motivation and related work for sensors selection/placement in SHM systems. A second challenge that arises when dealing with damage detection in structures is about how many sensors could be needed to detect a damage and how to place them. This topic has been widely investigated in the past years with different approaches. Theoretical algorithms based on Kalman filtering have been proposed in this sense outside the SHM community. For example, in Reference 24 the authors presented a greedy sensor selection algorithm based on sub modular functions, in Reference 25 the authors introduced a stochastic sensor selection algorithm based on a greedy algorithm, and in Reference 26 the authors proposed a solution to the problem of sensor selection for parameter estimation with correlated measurement noise and greedy algorithm. However, these methods are not contextualized in the SHM field. In this respect, much effort has been pushed into the use of information entropy for sensors selection/placement in SHM systems in the last 20 years. The first works in this sense were introduced in References 27,28 with the aim of minimizing the uncertainty of the estimated model parameters and exploiting the possibility to compare different configurations of sensors. Since then, lot of work has been done in this direction.

For example, in Reference 29 the authors discussed an entropy-based sensor placement based on the minimization of the information entropy. The problem is solved by an approximation of the information entropy estimation so that large number of data can be handled. Results also show that this kind of approach can outperform others, as for example the ones based on genetic algorithms. However, it is mainly based on heuristics and does not leverage any potential information that a modeling framework could provide, thus even modeling errors are neglected. Also, this approach is based on a single type of sensors, while other studies take into account the multi-type sensor configuration.

In Reference 30 the authors combined the Bayesian sequential sensor placement algorithm together with the concept of information entropy for the optimization of multi-type sensor configurations, however it only considered a single objective function and a single setup measurement. These issues are taken into account in the most recent papers,[31] where the authors consider an optimal sensor placement strategy for multi-setup modal testing of large structures, and References 32,33, where the authors provide different methodologies to address multi-objective optimization for sensor placement. Nonetheless,[31,33] do not consider a dynamical model of the structure that interacts with the sensor selection procedure, while Reference 32 only considers the equation of motion of a linear-elastic structural system and a specific damage detection model in the form of structural stiffness reduction.

Contribution. In this article we aim at addressing several of the mentioned limitations considering both challenges, that is, modeling and damage detection, and sensor selection/placement.

We propose two novel damage detection methodologies that are based on Kalman filtering. The first method exploits a recently developed model identification technique that is based on auto regressive (AR) system identification and regression trees (RTs) theory.[34-36] The main advantage of this methodology is that it provides a modeling framework that is both (1) simple from the computational complexity point of view (as it consists of a collection of linear models) and (2) accurate in terms of damage detection performance (as it will been shown in the validation section). Furthermore, thanks to the dataset partition induced by the RTs algorithm, it is also able to extract a piecewise linear model of the nonlinear behavior of a structure.

The second method consists of the extension of the aforementioned classical PCA modeling approach by means of the poly-exponential (PE) theory. PE is a well known modeling technique that has been used in several fields of study. For example, in References 37-39 the authors exploited PE-based methods to create body-drug reactions models in the study of pharmacokinetics. In Reference 40 the authors derived several identities connecting the PE functions and the degenerate Bell polynomials. In Reference 41 the authors modeled plasma concentration versus time profiles with the PE model. The proposed model was able to capture the time-course of drug concentrations with high accuracy. In Reference 42 the author applied the PE model to perform zero-noise extrapolation, where they were able to show error reductions of 18 to 24 fold over non-mitigated circuits. However, based on our literature review, it seems that this kind of modeling approach has never been introduced in the context of damage detection in SHM systems. Thus, it has motivated us to further dig the potential of this method. For this reason, we extended the classical PCA modeling framework by adding a PE (hence nonlinear) correction term, and set up a Kalman filter-based damage detection methodology, to show that such modeling extension does improve the damage detection sensitivity by capturing the nonlinear system behavior with respect to the classical PCA.

Last but not least, as a third contribution of this article we propose a novel entropy-based sensor selection methodology that integrates the proposed data-driven modeling of the structure. The goal is to use the concepts of entropy and information gain from Information theory to drastically reduce the number of sensors on a structure without affecting (and, in our experimental cases, even strongly improving) the predictive model accuracy. More precisely, the idea is to place n sensors over a structure (or leverage a structure that is already equipped with n sensors) with n quite large, with the objective of predicting the output of a specified sensor i, for example, identified from the structural engineer, and use them to collect a dataset. Then we iteratively apply our entropy-based methodology to this dataset to reduce the number of sensors to $n^* << n$. At the first cycle of the algorithm all possible pairs of sensors are considered, and the pair that maximizes the information gain is chosen. Then we iterate, considering all pairs except the chosen one. After n^* iterations all redundant sensors are removed so that they can be used with other structures to repeat the process, thus reducing the cost of long-term sensing campaigns. Moreover, although the proposed algorithm discusses about sensor selection, it is quite intuitive how it can be used in the problem of sensors placement from the point of view of damage detection. This is because it identifies the sensors set that provides non-redundant information between all the ones placed on the structure, and select the *useful* sensors locations. However, it does not allow the optimization of any cost function for the optimal sensors placement for applications other than damage detection, as is the case of part of the literature cited above, but this is out of the scope of the article. We also show in the simulation section that the proposed approach also scales well to different types of structures.

We validate our damage detection and sensor selection methodologies on three independent experimental datasets, that is, the bookshelf resembling structure, the Irvine bridge's column and the Sheraton hotel, provided from the repository of the Engineering Institute at the Los Alamos National Laboratory (LANL),[43] and show that they outperform the results obtained using well-established techniques as the plain PCA approach by Krishnan et al.[16] and the standard PLS approach over the same dataset.

This article extends preliminary results already published in Reference 44, where only the sensor selection algorithm and its application with the PCA on the bookshelf structure were proposed. In addition, in this article we extend the modeling with nonlinear components, that is, by including RTs and PE model structures, and validate the proposed methodologies on two additional experimental benchmarks.

Article organization. The article is organized as follows. In Section 2 we describe both the PCA-based technique used in Reference 16 and our novel model identification technique, based on AR identification and RTs theory, as well as the damage detection strategy. In Section 3 we illustrate our entropy-based novel sensor selection technique. In Section 4 we define the PE modeling and its integration with Entropy-based sensor selection methodology. In Sections 5.1–5.3 we validate the techniques derived in this article on 3 accelerometric datasets[3,43,45] provided by various institutions and accessible via the data repository of the LANL In particular, we compare our novel methodology with existing PCA-based methods both in terms of prediction accuracy and damage detection sensitivity.

2 | MODEL IDENTIFICATION AND DAMAGE DETECTION

In this section, we first describe two methods for model identification: the first based on PCA (illustrated in Reference 16), and the second based on a combination of RTs and AR model identification (as proposed in References 34-36). Then we introduce a novel methodology to use such models for damage detection via a Kalman filter-based residual generator.

2.1 | PCA-based damage detection

In Reference 16 the authors have demonstrated the capabilities of a PCA-based fault detection algorithm[46,47] for continuous online damage detection of vibrating structures, that we recall in this subsection. Their method consists of two main steps: (1) data projection onto an orthogonal subspace and AR terms identification, and (2) Kalman-based dynamic estimation of the corresponding AR coefficients.

Let us define $X = [\mathbf{x}_1\ \mathbf{x}_2 \cdots \mathbf{x}_n] \in \mathbb{R}^{m\times n}$ as the matrix of sensor measurements consisting of vectors $\mathbf{x}_i = [x_i(1)\ \cdots\ x_i(m)]^\top \in \mathbb{R}^m,\ i = 1, \dots, n$, where $x_i(k) \in \mathbb{R}$ denotes the measurement at time k of the ith sensor, n denotes the number of sensors (features), and m denotes the number of samples measured from each sensor. Denote by $R \in \mathbb{R}^{n\times n}$ the correlation matrix of X, and by $V \in \mathbb{R}^{n\times n}$ the matrix of eigenvectors of R sorted so that the corresponding eigenvalues of V are decreasing from the left to the right columns.

Step 1. The principal orthogonal component (POC) $\Psi \in \mathbb{R}^{m\times n}$ can be defined as

$$\Psi = XV, \tag{1}$$

and computed using the collected data. Then, for each jth PC ψ_j, that is, the column wise elements of Ψ, an AR model can be identified by

$$\psi_j(k) = \sum_{i=1}^{\tau} \alpha_{ij}^0 \psi_j(k-i), \tag{2}$$

where τ denotes the autoregressive order, and k is the discrete time variable evolving according to the sampling time. Coefficients α_{ij}^0 are computed via the classical least square method commonly used to identify AR models.

Step 2. Once the predictive model (2) is obtained, the next step is to define a residual generator for fault detection. Note that the AR coefficients in (2) refer to the system's nominal dynamics. The idea is to assume that the AR terms are subject to changes when for example a damage occurs. Thus, to estimate non-nominal changes of such AR terms, a state variable vector $\boldsymbol{\alpha}_j(k)$ for each component j of our system is defined, together with the associated dynamics

$$\begin{aligned} \boldsymbol{\alpha}_j(k+1) &= I\boldsymbol{\alpha}_j(k) + \omega(k), \\ \psi_j(k) &= \overline{\boldsymbol{\psi}}_j(k)\boldsymbol{\alpha}_j(k) + \nu(k), \end{aligned} \tag{3}$$

where $\boldsymbol{\alpha}_j(k) = [\alpha_{1j}(k), \dots, \alpha_{\tau j}(k)]$ is a state vector with initial condition $\boldsymbol{\alpha}_j(0) = \left[\alpha_{1j}^0, \dots, \alpha_{\tau j}^0\right]$, $\overline{\boldsymbol{\psi}}_j(k) = [\psi_j(k-1), \dots, \psi_j(k-\tau)]$ is a time varying observation matrix, $\omega(k)$ and $\nu(k)$ are uncorrelated white noise signals with variance

σ_ω^2 and σ_v^2 respectively, and I is the identity matrix of appropriate dimension. Since $\overline{\boldsymbol{\psi}}_j(k)$ and $\boldsymbol{\psi}_j(k)$ are given by sensor measurements transformed via (1), they are known at each time step k. In Reference 16 a Kalman filter can be easily constructed leveraging[48-50] to estimate the time evolution of the AR parameters. In particular, for each time step k, the Kalman filter solution for the model above is computed as

$$\begin{aligned}
&\boldsymbol{\alpha}_j(k|k-1) = \boldsymbol{\alpha}_j(k-1|k-1)\\
&\boldsymbol{P}_{\boldsymbol{\alpha}}(k|k-1) = \boldsymbol{P}_{\boldsymbol{\alpha}}(k-1|k-1) + I\cdot\sigma_\omega^2\\
&\psi_j(k|k-1) = \overline{\boldsymbol{\psi}}_j(k)\boldsymbol{\alpha}_j(k|k-1)\\
&S(k) = \overline{\boldsymbol{\psi}}_j(k)\boldsymbol{P}_{\boldsymbol{\alpha}}(k|k-1)\overline{\boldsymbol{\psi}}_j(k)^\top + \sigma_v^2\\
&\tilde{R}(k) = \psi_j(k) - \psi_j(k|k-1)\\
&K_G(k) = \boldsymbol{P}_{\boldsymbol{\alpha}}(k|k-1)\overline{\boldsymbol{\psi}}_j(k)^\top S(k)^{-1}\\
&\boldsymbol{\alpha}_j(k|k) = \boldsymbol{\alpha}_j(k|k-1) - K_G(k)\tilde{R}(k)\\
&\boldsymbol{P}_{\boldsymbol{\alpha}}(k|k) = [1 - K_G(k)\overline{\boldsymbol{\psi}}_j(k)]\boldsymbol{P}_{\boldsymbol{\alpha}}(k|k-1),
\end{aligned} \tag{4}$$

where $\boldsymbol{\alpha}_j(k|k-1)$ and $\boldsymbol{\alpha}_j(k|k)$ are the a priori and a posteriori estimates of $\boldsymbol{\alpha}_j$, $\boldsymbol{P}_{\boldsymbol{\alpha}}(k|k-1)$ and $\boldsymbol{P}_{\boldsymbol{\alpha}}(k|k)$ are the a priori and a posteriori estimates of the covariance, and $S(k)$, $\tilde{R}(k)$, $K_G(k)$ are respectively the pre-fit residual covariance, the prediction residual and the Kalman gain at instance k. The Kalman filter above provides an estimate of $\boldsymbol{\alpha}_j(k)$ in run-time: we will use such estimate to detect whether a damage/fault is running or not in the system, as will be explained in Section 2.3.

2.2 | RT-based damage detection

In References 34-36 the authors proposed a new model identification technique to derive a switching AR/ARX model starting from historical data of a system, leveraging standard model identification methods and RTs. To apply such identification technique in our context, we have to extend the dataset identified by the matrix X into an extended dataset $\mathcal{D}_x$ by adding the regressive terms of the measurements as features, that is,

$$\mathcal{D}_x = \{(x(k)^\top, x(k-1)^\top, \dots, x(k-\tau)^\top)\}_{k=1,\dots,m}, \tag{5}$$

where $x(k) = [x_1(k), \dots, x_n(k)]^\top \in \mathbb{R}^n$.

RTs are a technique from machine learning that partitions, via for example, the CART algorithm,[51] the historical dataset using specific rules. More precisely, the CART algorithm creates a RT structure via optimal partitioning of the dataset: it solves a least square problem by optimally choosing recursively a variable (feature) to split on and a corresponding splitting point. After several steps the algorithm converges to the optimal solution, and the dataset $\mathcal{D}_x$ in our case is partitioned in L hyper-rectangular sets R_ℓ such that $\bigcup_{\ell=1}^{L} R_\ell = \mathcal{D}_x$: each of these sets is associated to a leaf $\ell = 1, 2, \dots, L$ of the *learnt* tree $\mathcal{T}$. Then, in each partition we can estimate the value of the variable we wish to predict, in our case $x(k)$, with a constant given by the average of the samples in the partition (we refer to Reference 51 for more details regarding RTs).

In the novel method proposed in References 34-36, differently from CART, an AR model is fitted in each leaf ℓ using only the samples belonging to the corresponding hyper-rectangle R_ℓ:

$$x_j(k) = \sum_{i=1}^{\tau} \alpha^0_{ij,\ell(\overline{\mathbf{x}}(k))} x_j(k-i), \tag{6}$$

where $\overline{\mathbf{x}}(k) = [x(k-1)^\top \ \cdots \ x(k-\tau)^\top]$, and $\ell(\overline{\mathbf{x}}(k)) = \{\ell : \overline{\mathbf{x}}(k) \in R_\ell\}$ assigns to the state measurements $\overline{\mathbf{x}}(k)$ the corresponding leaf in $\mathcal{T}$.

We can now leverage this modeling to extend the previous section, defining for each component j a dynamical model of the AR terms

$$\begin{aligned}
&\boldsymbol{\alpha}_j(k+1) = I\boldsymbol{\alpha}_j(k) + \omega(k)\\
&x_j(k) = C(\overline{\mathbf{x}}(k))\boldsymbol{\alpha}_j(k) + \nu(k),
\end{aligned} \tag{7}$$

where

$$\boldsymbol{\alpha}_j(k) = [\alpha_{1j,1}(k) \ \ldots \ \alpha_{\tau j,1}(k) \ \ldots \ \alpha_{1j,L}(k) \ \ldots \ \alpha_{\tau j,L}(k)]^\top,$$

is a state vector with initial condition

$$\boldsymbol{\alpha}_j(0) = [\alpha^0_{1j,1} \ \ldots \ \alpha^0_{\tau j,1} \ \ldots \ \alpha^0_{1j,L} \ \ldots \ \alpha^0_{\tau j,L}],$$

$C(\overline{\mathbf{x}}(k))$ is a time varying observation matrix defined by

$$C(\overline{\mathbf{x}}(k)) = [\overline{\mathbf{x}}(k)\delta_{1,\ell(\overline{\mathbf{x}}(k))} \ \cdots \ \overline{\mathbf{x}}(k)\delta_{N,\ell(\overline{\mathbf{x}}(k))}],$$

with $\delta_{\cdot,\cdot}$ the Kronecker delta, $\omega(k)$ and $\nu(k)$ uncorrelated white noise signals with variance σ_ω and σ_ν respectively, and I the identity matrix of appropriate dimension. Note that in this case we are not using the PCA transformation as in (1), and $\overline{\mathbf{x}}(k)$ is directly measurable. With few modifications with respect to the previous section, a Kalman filter can be easily constructed leveraging[48-50] to estimate the time evolution of the AR parameters.

2.3 | Damage detection thresholds

To detect damages one can check whether the run-time trajectories of the $\boldsymbol{\alpha}_j(k)$ vectors in (3) or (7) change with respect to the nominal initial conditions, that is, whether the autoregressive coefficients change due to a change in the physics of the system. However, the choice of a threshold to decide whether the noisy change of $\boldsymbol{\alpha}(k)$ corresponds to a faulty event is in general application dependent.[13,52] In this article, without loss of generality, we define a scalar residual vector

$$\gamma(k) = ||\boldsymbol{\alpha}(k)||_2, \tag{8}$$

with $\boldsymbol{\alpha}(k) = [\boldsymbol{\alpha}_1(k), \ \ldots \ , \boldsymbol{\alpha}_n(k)]$, and bounds

$$\gamma_u = (1 + b\%)||\gamma(0)||_2, \quad \gamma_l = (1 - b\%)||\gamma(0)||_2. \tag{9}$$

We consider a system nominal when $\gamma(k)$ does not exceed the nominal value of more than $b\%$

$$\gamma_l \leq \gamma(k) \leq \gamma_u, \tag{10}$$

and faulty otherwise. The choice of $b\%$ purely depends on the application, and often heuristic methods are used in practice taking into account that large values of $b\%$ can avoid false alarms, but can either increase the time to detect a damage or even fail to detect it.

3 | ENTROPY-BASED SENSOR SELECTION

In this section, we are interested in reducing the number of sensors (accelerometers in our experimental setups) placed on a structure without losing (or even improving) the accuracy of the predictive models described in the previous section, and hence keeping (or even improving) the damage detection performance. In particular, the idea is to equip a structure with n sensors randomly placed all over it, or to consider a structure that is already equipped with n sensors, and use the collected measurement data to identify what sensors provide the needed (non-redundant) information to detect a damage on the structure. In this way, the "useless" sensors can be removed and used to repeat the process on other structures.

A naive method is by selecting subsets of accelerometers that are strongly correlated one to each other: to do so, the correlation matrix $R = [r_{ij}]^n_{i,j=1} \in \mathbb{R}^{n\times n}$ can be used, where $-1 \leq r_{ij} \leq 1$. Indeed, a well known methodology that is applied in practice is to group sensors corresponding to very large values of the coefficients r^2_{ij}, where $0 \leq r^2_{ij} \leq 1$ and r^2_{ij} is called coefficient of determination between the components i and j (see References 53,54 for further reading on this topic). In this article, we will specifically call this heuristic procedure *correlation-based sensor selection* (c-ss).

We propose in this article an alternative metric to perform the sensor selection based on information theory. In particular, the goal is to maximize the *information gain*. To this aim, it will be also necessary to introduce the concept of *entropy* from Information theory. For this reason, we call this sensor selection approach, whose theoretical foundation is provided in the following sections, *entropy-based sensor selection* (e-ss).

The advantage provided by the e-ss approach with respect to c-ss consists of selecting the sensors on the basis of a different criterion, that is, maximizing the information gain, as formally proved in Theorem 1. Beyond such theoretical validation, we illustrate in the simulation section how the e-ss approach provides better results in terms of damage detection with respect to c-ss over three different experimental real benchmarks.

3.1 | Background on information theory

We first briefly recall the notions of entropy and information gain from Information theory. Intuitively, the entropy represents the measure of uncertainty of a random variable, while the information gain represents the amount of information shared by 2 or more random variables (we refer to References 55-57 for further reading on the topic). Let Y be a random variable which takes values in the set $\{y_1, y_2, \dots, y_n\}$: the entropy of Y is defined as

$$H(Y) = -\sum_{i=1}^{n} \mathbb{P}(y_i) log_2(\mathbb{P}(y_i)), \tag{11}$$

where $\mathbb{P}(y_i)$ denotes the probability of Y to take the value of y_i. The value of entropy is non negative, and the larger the entropy, the larger the *uncertainty*. For example, entropy for a fair coin toss is 1, while for a 6 headed dice it is 2.585, which is larger than fair coin toss since there are more possible events to occur. When Y is a binary variable, that is, it takes value in {0,1}, and

$$Y = \begin{cases} 0, & \text{with probability } p \\ 1, & \text{with probability } 1-p \end{cases}$$

with $0 \leq p \leq 1$, its entropy is called *binary entropy* and is defined as

$$H(Y) = H(p) = -[p \cdot log_2(p) + (1-p) \cdot log_2(1-p)]. \tag{12}$$

Given two random variables X and Y, suppose that X is known. We define the conditional entropy of Y given X as

$$H(Y|X) = \sum_{x \in X} \mathbb{P}(X = x) H(Y|X = x)). \tag{13}$$

$H(Y|X)$ represents the uncertainty of Y given the information of X: $H(Y|X) = 0$ means that, for each value taken by X, we deterministically know the outcome of Y.

Finally, the information gain is a measure of mutuality between a pair of variables. Given two random variables X and Y, the information gain between them is defined as

$$I(Y;X) = H(Y) - H(Y|X). \tag{14}$$

As for entropy, the value of the information gain is non negative: this implies that $H(Y) \geq H(Y|X)$. However, as opposed to entropy, the larger the information gain the smaller the uncertainty.

3.2 | Entropy-based sensor selection

Let us consider as above a dataset described by the matrix $X \in \mathbb{R}^{m \times n}$ consisting of vector measurements $[\mathbf{x}_1\ \mathbf{x}_2 \cdots \mathbf{x}_n]$, and let $X_1, \dots, X_n$ be random variables corresponding to each component (sensor). Namely, each vector $\mathbf{x}_i =$

$[x_i(1)\cdots x_i(m)]^\top \in \mathbb{R}^{m\times 1}$ is a sequence of random outcomes of X_i, that is, the outcome of the random vector $\mathbf{X}_i \doteq [X_i \ \dots \ X_i]^\top$ consisting of m i.i.d. scalar random variables X_i.

As mentioned at the beginning, the e-ss algorithm illustrated in this section will be based on the notion of information gain, and consequently on the notion of entropy. Let us assume we wish to define a subset of sensors to predict the component i. To this aim, given a dataset described by a matrix X, we define a binary random variable Z_{ij}, $\forall j \neq i$, as

$$Z_{ij} = \begin{cases} 1 & \text{if } |X_i - \beta_{ij}X_j| < \theta \cdot \sigma(X_i) \\ 0 & \text{if otherwise} \end{cases}, \tag{15}$$

where $\theta \in [1, +\infty) \subset \mathbb{R}$, $\sigma(X_i)$ denotes the standard deviation of X_i and $\beta_{ij} = (\mathbf{x}_j^\top \mathbf{x}_j)^{-1}\mathbf{x}_j^\top \mathbf{x}_i$ is the solution of the least squares method. Let $p_{ij} = \mathbb{P}[Z_{ij} = 1]$. At this point, it is possible to define the information gain of Z_{ij} given β_{ij} as

where $\theta \in [1, +\infty) \subset \mathbb{R}$, $\sigma(X_i)$ denotes the standard deviation of X_i and $\beta_{ij} = (\mathbf{X}_j^\top \mathbf{X}_j)^{-1}\mathbf{X}_j^\top \mathbf{X}_i$ is a random variable defined as the solution of the least squares method. Let $p_{ij} = \mathbb{P}[Z_{ij} = 1]$. At this point, it is possible to define the information gain of Z_{ij} given β_{ij} as

$$I(Z_{ij}; \beta_{ij}) = H(Z_{ij}) - H(Z_{ij}|\beta_{ij}), \tag{16}$$

where $I(Z_{ij}; \beta_{ij})$ denotes the information gain of Z_{ij} given β_{ij}, that is, given $\mathbf{x}_i$ and $\mathbf{x}_j$.

The main idea is to estimate the information gain between each pair $i, j, j \neq i$, and chose the component j^* such that the information gain $I(Z_{ij}; \beta_{ij})$ is maximized. Since we do not know the expression of the information gain $I(Z_{ij}; \beta_{ij})$ and of the entropy $H(Z_{ij})$, the only way to proceed is by the conditional entropy $H(Z_{ij}|\beta_{ij})$ as in the following. Indeed, maximizing the information gain $I(Z_{ij}; \beta_{ij})$ is equivalent to choosing j^* that minimizes the conditional entropy $H(Z_{ij}|\beta_{ij})$. Although it is very difficult in general to compute exactly $H(Z_{ij}|\beta_{ij})$, as the main contribution of this section, we derive an upper bound $\hat{H}(Z_{ij}|\beta_{ij})$ of $H(Z_{ij}|\beta_{ij})$ as a function of the coefficient of determination r_{ij}^2, which can be computed easily. As a consequence, when we compute $\hat{j}^*$ that minimizes $\hat{H}(Z_{ij}|\beta_{ij})$ we are basically minimizing an upper bound of the conditional entropy, and thus maximizing a lower bound of the information gain. As widely experimentally validated in Sections 5.1–5.3, this approach strongly improves the prediction accuracy and the damage detection sensitivity with respect to the c-ss in all our three experimental setups.

In the following, with abuse of notation, we emphasize that the residual error and the coefficient of determination depend on the number of samples m using the notations $e_{ij}(m)$ and $r_{ij}(m)$.

Lemma 1. *Given Z_{ij} the following holds:*

$$p_{ij} \geq \lim_{m\to\infty} (1 - \theta^{-2} + \theta^{-2} \cdot r_{ij}^2(m)). \tag{17}$$

Proof. We recall that, for $m \to \infty$, $e_{ij}(m) = X_i - \beta_{ij}(m)X_j$ is a random variable representing the residual error and $\sigma(X_i)$ is the standard deviation of X_i. By the Chebyshev's inequality

$$\lim_{m\to\infty} \mathbb{P}(|e_{ij}(m) - \mu(e_{ij}(m))| \geq c\sigma(e_{ij}(m))) \leq \frac{1}{c^2}, \tag{18}$$

■

where $\mu(\cdot)$ and $\sigma(\cdot)$ respectively denote the expectation and the standard deviation, and c is any real positive number. We now show that, under the assumption of random sampling, $\mu(e_{ij}(m))$ and $\sigma(e_{ij}(m))$ can be well estimated from X_i when the number of samples m goes to infinity. Let $\mu_m(\cdot)$ and $\sigma_m(\cdot)$ respectively denote the sample mean and the sample standard deviation of the vector $\mathbf{x}_i$ of m samples, with

$$\lim_{m\to\infty} \mu_m(\mathbf{x}_i) = \mu(X_i), \ \lim_{m\to\infty} \sigma_m(\mathbf{x}_i) = \sigma(X_i). \tag{19}$$

We first note that $\mu_m(e_{ij}(m)) = 0$, $\forall m$ as a direct consequence of defining $\beta_{ij}(m)$ via the least squares algorithm. Then, the sample standard deviation of $e_{ij}(m)$ can be defined as

$$\sigma_m(e_{ij}(m)) = \sqrt{\frac{\sum_{k=1}^n (x_i(k) - \beta_{ij}(m)x_j(k))^2}{m - d}}, \tag{20}$$

where d is the degree of freedom. The coefficient of determination obtained from the dataset vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, each consisting of m samples, is defined as

$$r_{ij}^2(m) = 1 - \frac{\sum_{k=1}^{n}(x_i(k) - \beta_{ij}(m)x_j(k))^2}{\sum_{k=1}^{n}(x_i(k) - \mu_m(\mathbf{x}_i))^2}. \tag{21}$$

By simple manipulations on (20) and (21) it follows that

$$\sigma_m(e_{ij}(m)) = \sigma_m(\mathbf{x}_i)\sqrt{1 - r_{ij}^2(m)}. \tag{22}$$

We can now rewrite (18) as

$$\lim_{m\to\infty} \mathbb{P}\left(|e_{ij}(m)| \geq c\sigma_m(\mathbf{x}_i)\sqrt{1 - r_{ij}^2(m)} \right) \leq \frac{1}{c^2}. \tag{23}$$

By imposing $c = \frac{\theta}{\sqrt{1-r_{ij}^2(m)}} > 0$ we obtain

$$\lim_{m\to\infty} \mathbb{P}(|e_{ij}(m)| \geq \theta \cdot \sigma_m(\mathbf{x}_i)) \leq \lim_{m\to\infty} (\theta^{-2} - \theta^{-2} \cdot r_{ij}^2(m)). \tag{24}$$

By (19) if follows that

$$\lim_{m\to\infty} \mathbb{P}(|e_{ij}(m)| < \theta \cdot \sigma(X_i)) = p_{ij} \geq \lim_{m\to\infty} (1 - \theta^{-2} + \theta^{-2} \cdot r_{ij}^2(m)). \tag{25}$$

This concludes the proof.

Theorem 1. *Let*

$$\begin{aligned}
&\hat{H}(Z_{ij}|\beta_{ij}) \doteq \\
&-[(1 - \theta^{-2} + \theta^{-2} \cdot r_{ij}^2(m)) \log_2(1 - \theta^{-2} + \theta^{-2} \cdot r_{ij}^2(m)) \\
&+ (\theta^{-2} - \theta^{-2} \cdot r_{ij}^2(m)) \log_2(\theta^{-2} - \theta^{-2} \cdot r_{ij}^2(m))].
\end{aligned}$$

Then, for $m \to \infty$ *and* $r_{ij}^2 \in \left[1 - \frac{\theta^2}{2}, 1\right]$,

$$H(Z_{ij}|\beta_{ij}) \leq \hat{H}(Z_{ij}|\beta_{ij}).$$

Proof. $Z_{ij}|\beta_{ij}$ is a binary random variable, therefore

$$\begin{aligned}
H(Z_{ij}|\beta_{ij}) &= H(p_{ij}) \\
&= -[p_{ij} \log_2(p_{ij}) + (1 - p_{ij}) \log_2(1 - p_{ij})].
\end{aligned} \tag{26}$$

■

Since by Lemma 1 $p_{ij} \geq \lim_{m\to\infty}(1 - \theta^{-2} + \theta^{-2} \cdot r_{ij}^2(m))$, and by definition of the entropy function, it directly follows that $H(Z_{ij}|\beta_{ij}) \leq \hat{H}(Z_{ij}|\beta_{ij})$ for $(1 - \theta^{-2} + \theta^{-2} \cdot r_{ij}^2(m)) \in [0.5, 1]$, which directly implies the thesis since $\theta > 1$. This concludes the proof.

Remark 1. Note that when $r_{ij}^2 \in [0, 1 - \frac{\theta^2}{2})$ we cannot state any relation between $H(Z_{ij}|\beta_{ij})$ and $\hat{H}(Z_{ij}|\beta_{ij})$, namely we can just imply the trivial inequality $H(Z_{ij}|\beta_{ij}) < 1$ that holds by definition.

To validate Theorem 1 with a numerical test we generated 2 random vectors, namely x_i and g of 10^4 components, from normally distributed random variables X_i and G via the Matlab Random Number Generation (randn) function. Then, we defined the random variable $X_j = 2X_i + \xi G$. Thus, x_j is endowed with a linear correlation to x_i perturbed by ξg: the larger ξg the more independent x_j from x_i, and consequently the smaller r_{ij}^2. In particular, for the test we selected $\xi \in [0.75, 3.5]$.

TABLE 1 Numerical test spanning r_{ij}^2 for $\theta = 1, 1.05, 1.1$

$\theta = 1$			
r_{ij}^2	p_{ij}	$\hat{H}(Z_{ij}\|\beta_{ij})$	$H(Z_{ij}\|\beta_{ij})$
0.9613	1	0.2361	0
0.6051	0.913	0.9679	0.4264
0.4029	0.7514	0.9726	0.8091
0.3304	0.7089	0.9153	0.8701
0.3178*	0.698	0.9019	0.8837
0.2732	0.681	0.8461	0.9033
0.1639	0.6319	0.6436	0.9492
0.1409	0.6229	0.5865	0.956
0.1066	0.6102	0.4895	0.9647
$\theta = 1.05$			
0.9405	1	0.303	0
0.6355	0.9978	0.9156	0.0226
0.4941	0.8514	0.9951	0.6063
0.3011	0.722	0.9476	0.8527
0.2397*	0.6918	0.8936	0.8911
0.1938	0.6753	0.8397	0.9094
0.1117	0.6436	0.7103	0.9397
0.0952	0.6376	0.6786	0.9447
0.0712	0.6313	0.6285	0.9497
$\theta = 1.1$			
0.8782	1	0.4711	0
0.3124	0.7668	0.9865	0.7835
0.2257	0.7217	0.9428	0.8531
0.1687*	0.6966	0.8966	0.8854
0.1301	0.6804	0.8569	0.9039
0.103	0.6701	0.8247	0.9148
0.0834	0.6624	0.7991	0.9225
0.049	0.6495	0.7492	0.9345
0.0422	0.6467	0.7385	0.937

Note: * denotes the largest experimental r_{ij}^2 such that $\hat{H}(Z_{ij}|\beta_{ij}) \geq H(Z_{ij}|\beta_{ij})$ is still satisfied.

Starting from x_j and x_i, we calculated p_{ij} from (15) using Monte Carlo simulations, and $H(Z_{ij}|\beta_{ij})$ and $\hat{H}(Z_{ij}|\beta_{ij})$ as illustrated above in order to span from (almost) perfect linear correlation to (almost) no correlation at all. The results of the numerical test are shown in Table 1.

Using the above results we propose Algorithm 1 to choose, given a dataset described by the matrix X consisting of m traces of n sensors and a desired number of sensors $n^* < n$, a novel suboptimal and computationally fast sensor selection process based on information gain maximization. In the validation sections we compare the results obtained via our suboptimal sensor selection procedure with respect to the optimal solution, that we were able to compute via combinatorial search: clearly, this comparison is computationally possible only when the number of sensors under consideration, that is, n^* and n, is small.

Algorithm 1. Entropy-based subset selection

Inputs: Dataset described by matrix $X \in \mathbb{R}^{m\times n}$; Variable to predict $i \in \mathbf{n}$; Desired number of sensors $n^* < n$; Parameter θ.
Output: Set of sensors indices $S \subset \mathbf{n}$, $|S| = n^*$.
Initialization: $S := \{i\}$
for $k = 1, \dots, n^* - 1$ **do**
$j^* := \underset{j\in\mathbf{n}\setminus S}{argmin}\ \hat{H}(Z_{ij}|\beta_{ij})$
$S := S \cup \{j^*\}$
end for

Given any integer $n \in \mathbb{N}$, we denote by $\mathbf{n} = \{1, 2, \dots, n\}$ the set of integers up to n.

Remark 2. The complexity of the proposed algorithm is given by the sum of two terms: the first term is the complexity of computing the correlation matrix, and it is well known to be $\mathcal{O}(mn^2)$; the second term depends on the number of times that the *argmin* is computed at each cycle of the algorithm, considering that $\hat{H}(Z_{ij}|\beta_{ij})$ can be computed in constant time given r_{ij}. Thus, the algorithm complexity can be formalized as

$$\mathcal{O}(mn^2) + \sum_{k=1}^{n^*-1}(n-k)\mathcal{O}(1) = \mathcal{O}(mn^2) + \mathcal{O}(0.5(2n - n^*)(n^* - 1)) \approx \mathcal{O}(mn^2) + \mathcal{O}(nn^*) \approx \mathcal{O}(mn^2). \tag{27}$$

Therefore, the overall asymptotic complexity is the same of the c-ss algorithm, that is, the complexity of the computation of the correlation matrix.

Applying the e-ss algorithm to our experimental use cases in Sections 5.1–5.3 evidences the following tradeoff: when θ is chosen very close to 1 then clearly the e-ss and c-ss algorithms provide the same sensor selection in the range $r_{ij}^2 \in [1 - \frac{\theta^2}{2}, 1]$, while when θ is chosen much larger than 1 e-ss does not provide any benefit. In all our experimental datasets, the improvement of e-ss with respect to the c-ss occurs for values of θ around 1.05, namely when the random variable Z_{ij} represents the event that the prediction error on X_i given measurements of X_i, X_j does not exceed 5% of the variance of X_i. Given a specific dataset, the problem of solving the above tradeoff and choosing the optimal value of θ such that e-ss outperforms c-ss is an interesting venue for future work.

4 | POLY-EXPONENTIAL MODEL

In this section, we introduce a modeling approach based on the PE theory (Section 4.1), and use it as a nonlinear correction term to the ess-PCA (Section 4.2). In particular, we propose an idea to use such PE based correction term as an improvement to the e-ss PCA's model predictive accuracy, and to introduce a nonlinearity in the fault detector to improve fault detection performance as well.

4.1 | Poly-exponential correction term

Let us consider $\hat{X} = \Psi V^\top = [\hat{\mathbf{x}}_1\ \hat{\mathbf{x}}_2 \cdots \hat{\mathbf{x}}_n]$ as the estimate of X with the e-ss PCA algorithm, and $R = X - \hat{X} = [\mathbf{r}_1\ \mathbf{r}_2 \cdots \mathbf{r}_n]$ as the matrix of residual, with $\mathbf{r}_j = \mathbf{x}_j - \hat{\mathbf{x}}_j, j = 1, 2, \dots, n$, the residual vectors.

The goal is to fit a PE model $\hat{r}_j$, that is an approximation of the residual r_j, of the following form for each residual vector:

$$\hat{r}_j(k) = \sum_{i=1}^{p} b_{ij} e^{l_j(\hat{x}_j(k)-c_j)^i} + d_j, \tag{28}$$

where p is the order of the PE, and $b_{1j}, b_{2j}, \dots, b_{pj}, l_j, c_j$ and d_j are the model parameters that can be estimated in different ways, for example, with the Levenberg–Marquardt algorithm[58,59] as in our case. As a consequence, given (28), a new

model estimate of $\mathbf{x}_j$, provided by the PE model and denoted by $\tilde{\mathbf{x}}_j$, is defined as

$$\tilde{x}_j(k) = \hat{x}_j(k) + \sum_{i=1}^{p} b_{ij} e^{l_j(\hat{x}_j(k)-c_j)^i} + d_j. \tag{29}$$

As in the previous section, we can extend (28) to implement the fault detection algorithm. As before, we assume that the nonlinear regressive terms in (28) remain constant under nominal condition. Let

$$\boldsymbol{\beta_j} = [b_{1j}, b_{2j}, \ \dots \ , b_{pj}, l_j, c_j, d_j],$$

be the state vector, then the dynamical model for (28) can be defined as

$$\begin{aligned} \boldsymbol{\beta}_j(k+1) &= I\boldsymbol{\beta}_j(k) + \omega(k) \\ r_j(k) &= \sum_{i=1}^{p} b_{ij}(k) e^{l_j(k)(\hat{x}_j(k)-c_j(k))^i} + d_j(k) + \nu(k), \end{aligned} \tag{30}$$

where $\omega(k)$ and $\nu(k)$ are uncorrelated white noise signals with variance σ_ω^2 and σ_ν^2 respectively.

Finally, the time evolution of the nonlinear parameters in (30) can be estimated leveraging the extended Kalman filter, whose solution is expressed as

$$\begin{aligned} &\boldsymbol{\beta}_j(k|k-1) = \boldsymbol{\beta}_j(k-1|k-1) \\ &\boldsymbol{P}_\beta(k|k-1) = \boldsymbol{P}_\beta(k-1|k-1) + I \cdot \sigma_w^2 \\ &r_j(k|k-1) = \sum_{i=1}^{p} b_{ij}(k|k-1) e^{l_j(k|k-1)(\hat{x}_j(k)-c_j(k|k-1))^i} \\ &\qquad\qquad + d_j(k|k-1) \\ &\boldsymbol{J}(k) = \left.\frac{\partial r_j(k|k-1)}{\partial \boldsymbol{\beta}_j}\right|_{\beta(k|k-1)} \\ &S(k) = \boldsymbol{J}(k)\boldsymbol{P}_\beta(k|k-1)\boldsymbol{J}(k)^\top + \sigma_\nu^2 \\ &\tilde{R}(k) = r(k) - r(k|k-1) \\ &K_G(k) = \boldsymbol{P}_\beta(k|k-1)\boldsymbol{J}(k)^\top S(k)^{-1} \\ &\boldsymbol{\beta}_j(k|k) = \boldsymbol{\beta}_j(k|k-1) - K_G(k)\tilde{R}(k) \\ &\boldsymbol{P}_\beta(k|k) = [1 - K_G(k)\boldsymbol{J}(k)]P_\beta(k|k-1), \end{aligned} \tag{31}$$

where $\boldsymbol{\beta}_j(k|k-1)$ and $\boldsymbol{\beta}_j(k|k)$ are the a priori and a posteriori estimates of $\boldsymbol{\beta}_j$, $\boldsymbol{P}_\beta(k|k-1)$, and $\boldsymbol{P}_\beta(k|k)$ are the a priori and a posteriori estimates of the covariance, and $S(k)$, $\tilde{R}(k)$, $K_G(k)$, $\mathbf{J}(k)$ are the pre-fit residual covariance, the prediction residual, the Kalman gain and the Jacobian vector, respectively. The extended Kalman filter solution above gives an estimate of $\boldsymbol{\beta}_j(k)$ in the run-time, and we can use it to detect whether a fault existed or not in the system.

However, other than using such modeling as it is, the idea is to use it as a nonlinear correction term for e-ss PCA approach. We show in Section 5.4 how the combination of these 2 approaches provides a performance improvement in terms of fault detection.

4.2 | E-ss PCA + PE correction term

Let us consider the definitions already provided in Section 4.1, together with (29) and (2). Let also recall that

$$\begin{aligned} \hat{x}_j(k) &= \sum_{i=1}^{n} \psi_i(k) v_i(j) \\ &= \sum_{i=1}^{n} \sum_{t=1}^{\tau} \alpha_{ti} \psi_i(k-t) v_i(j). \end{aligned} \tag{32}$$

Then, by substituting (32) into (29) we get

$$\tilde{x}_j(k) = \sum_{i=1}^{n}\sum_{t=1}^{\tau}\alpha_{ti}\psi_i(k-t)v_i(j) + \sum_{i=1}^{p} b_{ij}e^{l_j\left(\sum_{i=1}^{n}\sum_{t=1}^{\tau}\alpha_{ti}\psi_i(k-t)v_i(j)-c_j\right)^i} + d_j. \tag{33}$$

Let now the new state vector be defined as

$$\boldsymbol{\gamma_j} = [\alpha_{11}, \alpha_{12}, \ldots, \alpha_{1\tau}, \alpha_{21}, \ldots, \alpha_{2\tau}, \ldots, \alpha_{n\tau}, b_{1j}, b_{2j}, \ldots, b_{pj}, l_j, c_j, d_j], \tag{34}$$

then the dynamical model for (33) can be defined as

$$\begin{aligned}\boldsymbol{\gamma}_j(k+1) &= I\boldsymbol{\gamma}_j(k) + \omega(k),\\ \tilde{x}_j(k) &= \sum_{i=1}^{n}\sum_{t=1}^{\tau}\alpha_{ti}(k)\psi_i(k-t)v_i(j)\\ &\quad + \sum_{i=1}^{p} b_{ij}(k)e^{l_j(k)\left(\sum_{i=1}^{n}\sum_{t=1}^{\tau}\alpha_{ti}\psi_i(k-t)v_i(j)-c_j(k)\right)^i} + d_j(k) + \nu(k).\end{aligned} \tag{35}$$

Remark 3. It is worth of notice that $\boldsymbol{\gamma}_j \in \mathbb{R}^{n\tau+p+3}$, that means there are $n\tau + p + 3$ equations to be derived for the Jacobian matrix (i.e., there are 200 variables for the bookshelf resembling structure, in our use case), which makes the extended Kalman filter solution for the Equation (35) to be computationally expensive. However, an approximation can be considered: since the α coefficients are independent from the β ones, their variation through the Kalman filter solution can be independently obtained via (3); thus, the variation of the β coefficients is computed via (30) given the αs.

Remark 4. This approximation is close to the real one especially when the first eigenvalue is large enough, such that the first PC contains the most information, characterized by high variance (i.e., the other PCs can be dropped due to insignificance).

In the next section we show that indeed considering both $\boldsymbol{\alpha_j}$ and $\boldsymbol{\beta_j}$ together does improve the capability of the model in detecting faults with respect to PCA and PE only.

5 | VALIDATION

This section is organized in two main parts:

1. In the first one we show the capability of the e-ss PCA and RT-based approaches in detecting faults. The validations, respectively addressed in Sections 5.1–5.3, are done on three different datasets, that is, the bookshelf resembling structure, the Irvine bridge's column and the Sheraton hotel, which are provided by the LANL in their online data repository.[45] In particular, we show that the e-ss version of both algorithms outperforms the standard and the c-ss versions of the respective algorithms. Furthermore, we show that the RT-based approach, especially the e-ss version, is more sensitive in detecting faults compared to the other techniques;
2. In the second part, we show the performance of the e-ss PCA with the PE correction term. We validate, in Section 5.4, its performance strictly on the e-ss version of PCA algorithm, as it is the one that provides the best performance. We also compare the performance of the e-ss PCA + PE to a standard learning algorithm, that is, the PLS, to show that a simple algorithm might not be good to predict faults, thus emphasizing that our work is indeed relevant and takes into account necessary complexity to detect faults in complex structures.

   We want to remark that the PLS algorithm can be used to detect faults following the same approach defined in the previous sections for the PCA-based algorithm via Kalman filtering. We do not explicitly describe the PLS-based fault detection algorithm only to avoid repetitions.

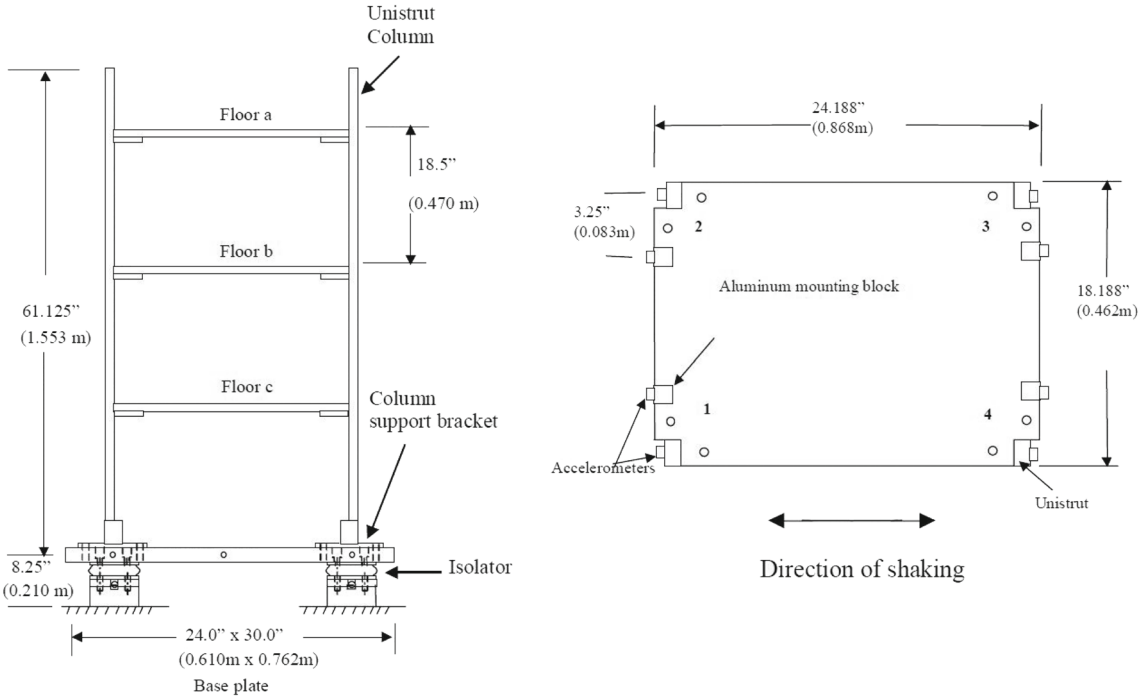

FIGURE 1 Assembled LANL test structure similar to a 3-tier bookshelf

5.1 | Bookshelf resembling structure use-case

This accelerometric dataset is provided from the repository of the Engineering Institute at the LANL.[3] The frame structure, which resembles a three tiers bookshelf, was built from unistrut columns and aluminum plates. Each of the shelves, which is 1.3 cm thick aluminum plate, is mounted on the unistrut column with 2-bolts connection brackets. In each tier 8 accelerometers (2 per joint) were installed on the plate and column. Figure 1 illustrates the structure and accelerometers placement for each tier. To generate the data, the structure was vibrated with frequency ranging in 0–3 KHz for 8 s, producing 4096 data samples. Several experiments in the nominal configuration were repeated, generating nominal (or baseline) datasets. Then, three different types of damage were induced in the structure: each of them is characterized by a different location of the damage, that is, in case 1 a damage was induced on joint 2a, in case 2 a damage was induced on joint 4b, and in case 3 a damage was induced both on joints 2a and 4b. To verify our fault detection algorithms capability to detect damages we will consider all 3 cases of damage provided by LANL. In particular, we will train our models with 3000 samples of the nominal behavior. We will use 1000 samples of the nominal behavior and 4000 samples of each of the 3 damages to validate respectively the prediction accuracy, and the damage detection sensitivity of our algorithms, comparing them with previous work on the same dataset.[16]

5.1.1 | Predictive model accuracy

We validate the prediction accuracy of the PCA- and RT-based models in 3 cases: (1) using all sensor components; (2) using sensors selected via c-ss; (3) using sensors selected via e-ss. We considered a selection of sets of sensors with cardinality $n^* = 5$ as it happens to be the best tradeoff between prediction accuracy and computational complexity. Of course, this choice can be adjusted to suit the specific application needs. The prediction accuracy will be measured using the normalized root mean square error (NRMSE).

Tables 2 and 3 summarize the accuracy of the PCA-based and RT-based predictive models respectively, where the ith row of each table indicates (1 - NRMSE) ×100% for predicting sensor component i using respectively all sensors, the sensor subset obtained using the c-ss and the sensor subset obtained using the e-ss. The last row indicates the average prediction accuracy among all 24 components. The last column of Table 2 refers to our "Oracle," namely the optimal sensor selection (opt-ss) obtained with a combinatorial search over all subsets of sensors to obtain the set that provides

TABLE 2 The results in the table are shown as normalized root mean square error in percentage, that is, (1 - NRMSE) ×100%

all PCA[16]	c-ss PCA	e-ss PCA	opt-ss PCA
91.88	95.36*	96.35*	97.87
91.03	92.65	95.07*	97.08
89.82	92.50	95.04*	97.21
91.18	93.15	95.95*	97.10
92.20	94.71*	94.94*	97.78
90.87	92.53	94.87*	97.12
89.90	92.61	95.20*	97.11
90.48	92.17	95.29*	96.46
93.25	95.60	96.50*	96.93
94.96	96.69*	96.42*	98.22
91.94	94.46	95.55*	96.24
93.15	93.97*	92.18	97.20
92.67	95.15	96.46*	97.41
95.35	96.37*	96.01*	98.38
91.44	94.24*	94.99*	95.71
93.59	94.14*	92.61	97.31
90.40	94.04	95.24*	95.65
91.16	93.02	95.09*	96.57
95.62	95.94*	95.98*	96.79
96.18	96.73*	96.21*	97.35
92.36	93.98	95.18*	95.88
90.99	94.75*	94.88*	96.65
95.42	95.65*	96.22*	96.74
95.80	96.16*	95.47*	97.33
92.57	**94.44**	**95.32***	**96.96**

Note: The last row indicates the average prediction accuracy among all 24 components. Results are about sensor selection sets for the PCA-based model. * denotes the best accuracy among all, c-ss and e-ss. Results show that both c-ss and e-ss PCA algorithms always achieve better prediction accuracy with respect to the case where all 24 sensors are used. Also, in most cases e-ss outperforms c-ss, and in general is better average-wise, thus it is a valid alternative w.r.t. the c-ss.

the best achievable prediction accuracy: although computationally very expensive, with just 24 sensors it was possible to compare all possible subsets, while in the case of a much larger number of sensors this would have not been possible because computationally intractable.

Table 2 shows that both c-ss and e-ss PCA algorithms always achieve better prediction accuracy with respect to using all 24 sensors. This result is somehow related to the linearity problem in PCA: indeed, using PCA each PC needs to be written as a linear combination of the original variables, but if the linearity between original variables does not exist PCA will not work properly. By introducing sensor selection into the standard PCA algorithm the linearity in the model is maintained by choosing the more linearly correlated components, thus increasing the prediction accuracy. In particular, the accuracy of e-ss PCA is larger than that of c-ss in almost all cases (up to +3% in accuracy) and in the average (+1% in accuracy, reducing the optimality gap of 50%): as widely discussed in the introduction, our interpretation is that the e-ss is indeed able to select the components that add more information for the predictive model. With respect to the computational time, e-ss PCA is characterized by more steps compared to the standard PCA and c-ss PCA. However these extra steps, which consist of the calculation of the approximate entropy followed by finding the minimum of it, are solvable within time complexity of $O(n)$:[60] therefore, e-ss improves the prediction accuracy and does not increase significantly the computational complexity.

TABLE 3 The results in the table are shown as normalized root mean square error in percentage, that is, (1 - NRMSE) ×100%

all RT[16]	c-ss RT	e-ss RT
92.03*	91.77	92.06
93.16*	92.6	91.08
91.37*	91.26	90.28
92.37*	92.29	90.76
92.8*	92.08	91.58
92.89	92.6*	91.11
91.99*	91.34	90.92
92.16*	92.02	91.13
91.58	91.67	92.07*
91.69	92.18*	91.7*
90.54	90.69*	90.32
91.78	91.34	92.07*
92.39	91.64	92.48*
92.46*	91.6	92.24
90.21	90.62	90.8*
91.21	91.4*	91.13
90.51*	90.15	90.43
91.13	91.31*	91.17
91.79*	91.79*	92.43
91.43	91.49	91.73*
90.81*	90.26	90.5
91.43	91.56*	90.91
91.93	92.02*	91.26
91.82	91.61	91.86*
91.73*	**91.55**	**91.33**

Note: The last row indicates the average prediction accuracy among all 24 components. Results are about sensor selection sets for the RT-based model. * denotes the best accuracy among all, c-ss and e-ss. Results show worse results with respect to PCA-based models, probably due to the inherent linearity of the dataset dynamics which can be better captured by the linear models of PCA than the nonlinear models derived exploiting the RT-based methodology. Moreover, there is no significant improvement in terms of prediction accuracy between the three methods, although e-ss and c-ss use significantly less sensors (5 instead of 24) to provide the same level of accuracy.

Table 3 shows that RT-based models provide worse results with respect to PCA-based models, probably due to the inherent linearity of the dataset dynamics which can be better captured by the linear models of PCA than the nonlinear models derived exploiting the RT-based methodology. Moreover, there is no significant improvement of the prediction accuracy using e-ss RT and c-ss RT with respect to all RT: of course, both e-ss and c-ss use significantly less sensors (5 instead of 24) to provide the same level of accuracy.

5.1.2 | Damage detection sensitivity

In this section we compare the sensitivity of damage detection algorithms using the predictive models introduced in this article, including the one proposed in Reference 16 that exploits all sensors. As discussed above we consider 3 datasets provided by LANL, each consisting of 4000 samples, corresponding to 3 different damages. Following the methodology described in Section 2, we first derive models on the nominal dataset and then apply the Kalman filter to generate a

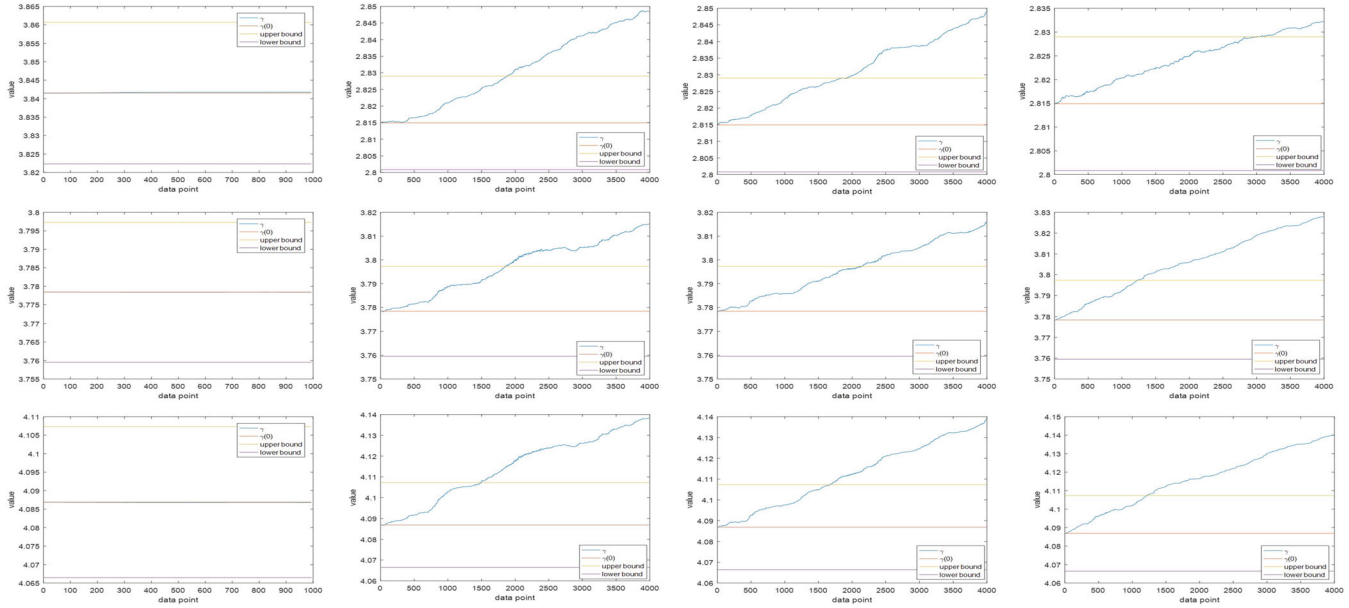

FIGURE 2 Trajectories of $\gamma(k)$: 1st column: nominal case, 2nd–4th column: faulty cases 1–3; 1st row: all PCA, 2nd row: c-ss PCA, 3rd Row: e-ss PCA

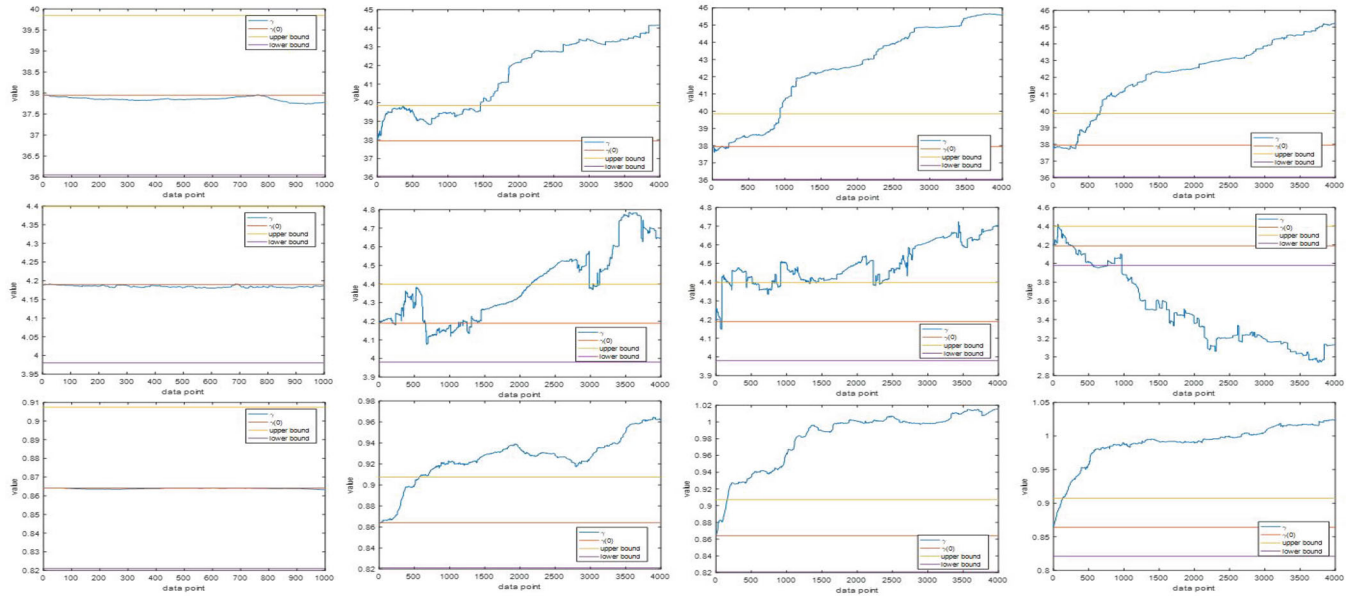

FIGURE 3 Trajectories of $\gamma(k)$: 1st column: nominal case, 2nd–4th column: faulty cases 1–3; 1st row: all RT, 2nd row: c-ss RT, 3rd row: e-ss RT

residual signal $\gamma(k)$: when such signal violates some predefined bounds, we raise a damage detection. In our simulations we have chosen $b = 0.5\%$ after a trial-and-error calibration process.

Figures 2 and 3 plot the trajectories of $\gamma(k)$ in nominal and faulty cases using respectively PCA-based and RT-based predictive models as described above, and simulations show that damages are detected by all methods.

To compare the damage detection sensitivity we define the *escape time* t_e as the minimum time instant such that the residual signal $\gamma(k)$ leaves the *nominal zone* bounded by γ_l, γ_u, and never returns back again, that is,

$$t_e \doteq \min\{t^* \geq 0 : (\forall t \geq t^*, \gamma(t) \notin [\gamma_l, \gamma_u])\}.$$

TABLE 4 Escape time t_e for different damages and techniques

Damage	all PCA	c-ss PCA	e-ss PCA	all RT	c-ss RT	e-ss RT
1	3.68	3.51	2.93	3.02	6.31	1.23
2	3.74	4.16	3.19	1.96	4.90	0.51
3	5.73	2.41	2.37	1.28	2.17	0.50

Note: Results show that the e-ss version for both PCA- and RT-based models performs better than c-ss, and in particular the e-ss PCA and e-ss RT reduce the escape time w.r.t. the c-ss PCA up to 23% and 88% respectively.

Table 4 clearly shows that e-ss version for both PCA- and RT-based models performs better than c-ss. More precisely, e-ss PCA and e-ss RT reduce the escape time w.r.t. c-ss PCA up to 23% and 88% respectively. It is interesting to note that e-ss RT, whose prediction accuracy is smaller than the PCA-based methods, outperforms all other methods in terms of damage detection sensitivity: our interpretation of this fact is that the nonlinear dynamics of the RT-based model is less accurate but more responsive to sudden changes in the structure.

5.2 | UCI bridge column use-case

This dataset is taken from an experimental setup performed by University of California, Irvine (UCI) on a bridge column.[43] The structure consists of two concrete bridge columns with diameter of 0.61 m retrofitted to columns of diameter 0.91 m. Both columns are 3.45 m in length and cast on the top of concrete foundations 0.635 m tall, and 40 accelerometers are assigned to the structures to measure the dynamics of the columns. Figure 4 illustrates the structure and the accelerometers placement on the bridge's column. The structure was vibrated by a shaker with frequency between 0 and 400 Hz for 8 s, and produced 2048 samples. The experiment was repeated twice, thus creating 2 nominal datasets. Faulty datasets, that is, the datasets containing data collected from the damaged structure, were generated with the same procedure as above. More precisely, an hydraulic actuator was used to apply lateral loads to the top of the column before the vibration process. The loads were applied in a force-controlled manner to produce lateral deformations at the top of the column corresponding to $0.15\Delta y$, $0.25\Delta y$, $0.4\Delta y$, and $0.7\Delta y$, where Δy denotes the deformation with respect to the y axis of the column. These experiments were repeated multiple times generating 8 faulty datasets containing 2048 samples each. To verify our fault detection algorithms capability to detect damages we considered 3 cases of damage provided by LANL: they consist on $0.25\Delta y$, $0.4\Delta y$, and $0.7\Delta y$ lateral deformation as in Reference 43. We trained our models using 2000 samples of the nominal behavior (1000 samples from each nominal dataset). We used the remaining 2096 samples from the nominal behavior (1048 samples from each dataset) to validate the prediction accuracy of our models. Finally, for the damage detection sensitivity of our algorithms we used 2048 samples from each of the 3 faulty datasets. We considered the datasets with $0.25\Delta y$ as damage level 1, which is the lowest, $0.4\Delta y$ as damage level 2, and $0.7\Delta y$ as damage level 3 (the higher the level, the stronger the damage).

5.2.1 | Predictive model accuracy

We proceed as illustrated above, considering a selection of sets of sensors with cardinality $n^* = 5$. Tables 5 and 6 summarize the accuracy of the PCA-based and RT-based predictive models respectively. As above, the last row indicates the average prediction accuracy among all 40 components.

Table 5 shows that both c-ss and e-ss PCA algorithms always achieve better prediction accuracy with respect to using all 40 sensors. Also in this case, the accuracy of e-ss PCA is larger or almost equal to c-ss in almost all cases (up to +14% in accuracy) and in the average (+3.5% in accuracy, reducing the optimality gap of 50%).

Table 6 reveals that, also in this case, the RT-based nonlinear models perform worse than the PCA-based linear models.

5.2.2 | Damage detection sensitivity

In this section we compare the sensitivity of damage detection algorithms using the predictive models introduced in this article. Following the methodology described in Section 2, we first derive models on the nominal dataset, and then apply

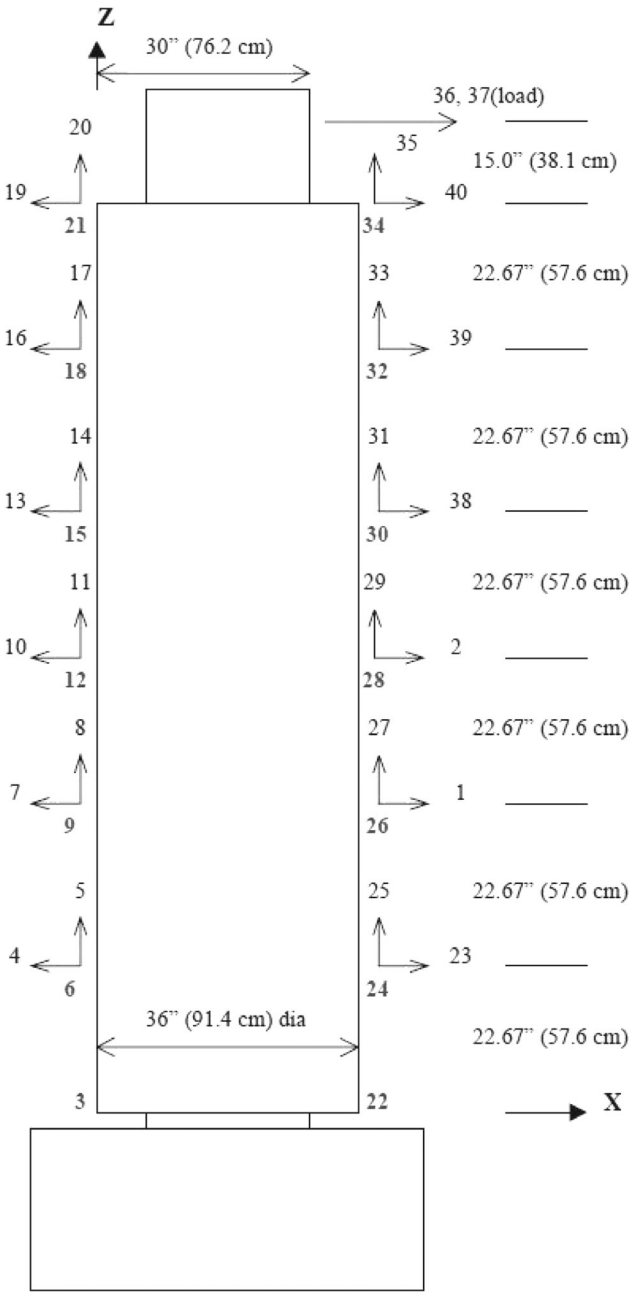

FIGURE 4 Accelerometers location on UCI bridge column

the Kalman filter to generate a residual signal $\gamma(k)$: when such signal violates some predefined bounds, we raise a damage detection. In this use-case we have chosen $b = 5\%$ after a trial-and-error calibration process.

Figures 5 and 6 plot the trajectories of $\gamma(k)$ in nominal and faulty cases using respectively PCA-based and RT-based predictive models as described above, and simulations show that damages are detected by almost all methods.

Table 7 clearly shows that, for Damages 2 and 3 (the strongest damages), all methods work well, without substantial differences. Regarding Damage 1 (the weakest damage and therefore the most difficult to be detected), c-ss PCA fails to detect the damage, all RT and c-ss RT are characterized by a very large escape time, while all PCA, e-ss PCA, and e-ss RT are characterized by a similar and small escape time. Since e-ss PCA and e-ss RT use only 5 sensors and all PCA uses 40 sensors, we can state that the e-ss PCA and e-ss RT outperform all other methods in detecting the weak Damage 1. This is crucial as, very often, strong damages in structures start from weak damages: being able to detect subtle damages is a huge advantage to prevent the occurrence of strong damages and catastrophic events.

TABLE 5 The results in the table are shown as normalized root mean square error in percentage, that is, (1 - NRMSE) ×100%

all PCA[16]	c-ss PCA	e-ss PCA	opt-ss PCA	all/e-ss PCA
86.04	94.21*	92.75	96.02	94.32
90.12*	90.5*	90.16*	93.14	N/A
83.23	87.07	90.38*	94.57	87.53
74.31	93.01*	92.63*	95.2	91.66
81.38	82.99*	82.31*	93.19	84.29
81.95	93.38	94.5*	95.62	92.72
90.08	94.39*	94.99*	96.31	94.03
18.09	74.94	81.82*	91.84	80.69
91.84	96.09*	95.84*	96.12	94.18
88.6	95.82*	94.81	96.68	95.29
20.46	82.04	89.28*	92.64	89.05
94.64	96.77*	93.65	97.15	96.74
89.51	95.96*	95.11*	96.89	95.76
63.95	85.26	86.42*	91.77	85.62
83.84	95.64*	95.74*	97.04	95.37
91.45	94.66	95.45*	96.6	94.64
55.5	75.48	89.56*	92.02	81.97
93.9	95.24	96.68*	97.15	95.88
86.08	94.17*	93.55*	96.4	92.7
34.87	76.76	88.3*	89.67	85.42
91.79	93.81*	92.69	97.09	94.51
77.86	86.5*	86.5*	94.47	86.69
77.34	92.5*	92.13*	94.53	91.16
90.63	92.79*	91.56	95.11	91.07
24.44	78.58*	77.74*	92.4	80.8
92.65	94.77*	94.77*	96.21	94.71
63.29	87.88*	87.76*	93.25	87.02
92.5	96.25*	96.27*	97.01	95.59
70.63	92.04*	85.98	94.26	89.24
96.08	97.35*	95.75	97.52	96.87
84.57	93.4*	92.65*	93.93	92.11
94.89*	95.99*	93.72	97.15	96.1
70	91.42*	89.22	93.99	91.04
93.41*	93.63*	93.63*	96.46	94.37
40.68	86.66	92.38*	94.23	89.47
86.34	90.38*	89.79*	92.93	90.51
91.95*	89.9	90.89*	93.83	N/A
85.95	95.37*	95.01*	96.74	95.76
85.58	88.24*	86.6	91.27	N/A
87.95	88.3*	88.66*	92.14	N/A
77.46	**89.42**	**91.05**	**94.76**	**91.25**

Note: The last row indicates the average prediction accuracy among all 24 components. Results are about sensor selection sets for the PCA-based model. * denotes the best accuracy among all, c-ss and e-ss. Results show that both c-ss and e-ss PCA algorithms always achieve better prediction accuracy with respect to the use of all 40 sensors. In particular, the accuracy of e-ss PCA is larger or almost equal to c-ss in almost all cases.

TABLE 6 The results in the table are shown as normalized root mean square error in percentage, that is, (1 - NRMSE) ×100%

all RT	c-ss RT	e-ss RT	all/ess RT
87.77	91.9*	91.42*	87.77
89.92	92.28*	92.27*	N/A
87.41	90.97*	90.97*	87.41
87.04	90.94*	90.75*	88.14
88.23	91.37*	91.92*	88.23
87.84	91.78*	91.36*	87.84
88.7	92.88*	92.83*	88.7
87.96	90.64*	90.97*	87.96
88.77	91.93*	91.84*	88.77
88.62	93.39*	93.66*	88.62
89.17	91.74*	91.74*	89.17
89.58	92.8*	92.8*	89.58
89.45	94.56*	94.47*	89.45
87.81	90.14*	89.9*	87.81
89.2	93.02*	93.42*	89.2
89.23	94.57*	94.64*	89.23
87.97	90.49*	90.25*	88.74
88.49	92.7*	93.54*	88.49
90.79	94.28*	94.06*	90.74
86.01	88.54*	87.71*	86.01
88.48	92.61*	92.7*	88.48
87.43	91.09*	91.09*	87.43
87.92	91.35*	90.8*	88.86
87.31	91.11*	90.65*	87.31
88.11	90.69*	90.7*	88.11
85.95	90.86*	90.86*	85.95
88.14	91.02*	90.97*	89.32
89.07	91.37*	91.7*	89.07
87.77	91.32*	91.61*	87.77
89.43	92.45*	92.41*	89.43
88.87	91.84*	92.4*	88.87
88.58	92.36*	92.36*	89.91
88.97	92.11*	92.2*	90
88.64	92.81*	92.81*	88.64
90.15	92.28*	92.81*	90.32
88.03	91.35*	91.43*	88.99
88.87	91.84*	91.84*	N/A
89.09	94.58*	94.47*	89.09
87.68	90.25*	90.26*	N/A
89.03	91.52*	91.47	N/A
88.43	**91.83***	**91.70***	**88.59**

Note: The last row indicates the average prediction accuracy among all 24 components. Results are about sensor selection sets for the RT-based model. * denotes the best accuracy among all, c-ss and e-ss. Sensor selection sets are same as Table 5. Results show that the RT-based nonlinear models perform worse than the PCA-based linear models.

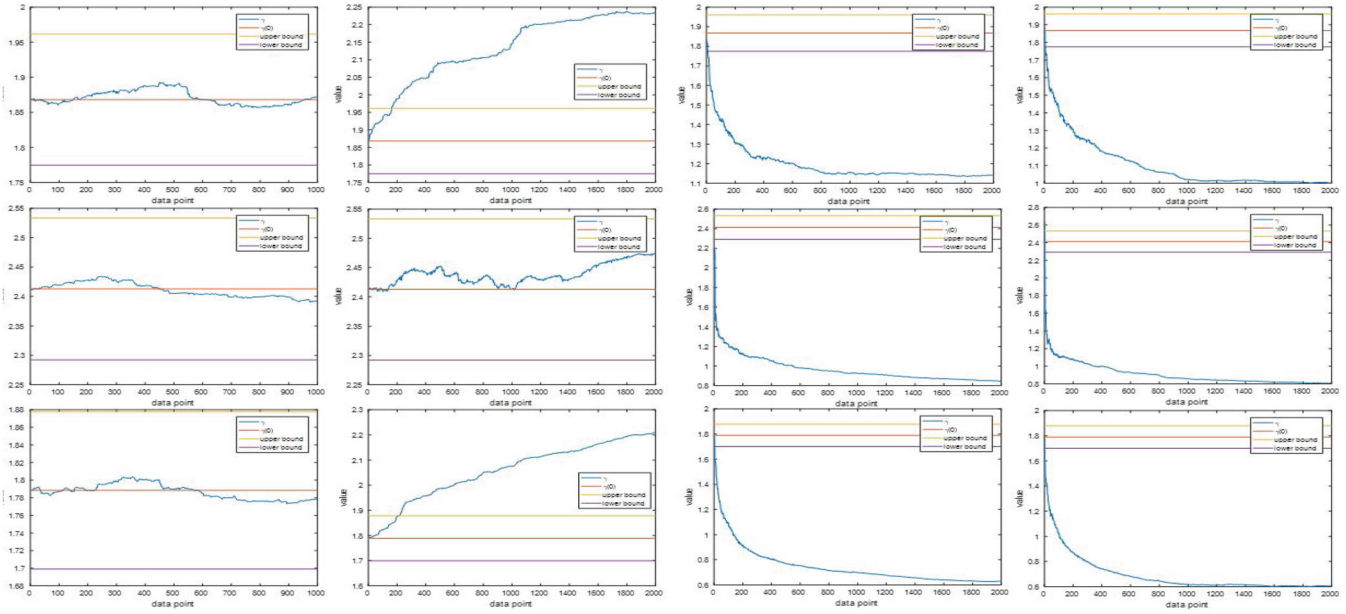

FIGURE 5 Trajectories of $\gamma(k)$: 1st column: nominal case, 2nd–4th column: faulty cases 1–3; 1st row: all PCA, 2nd row: c-ss PCA, 3rd row: e-ss PCA

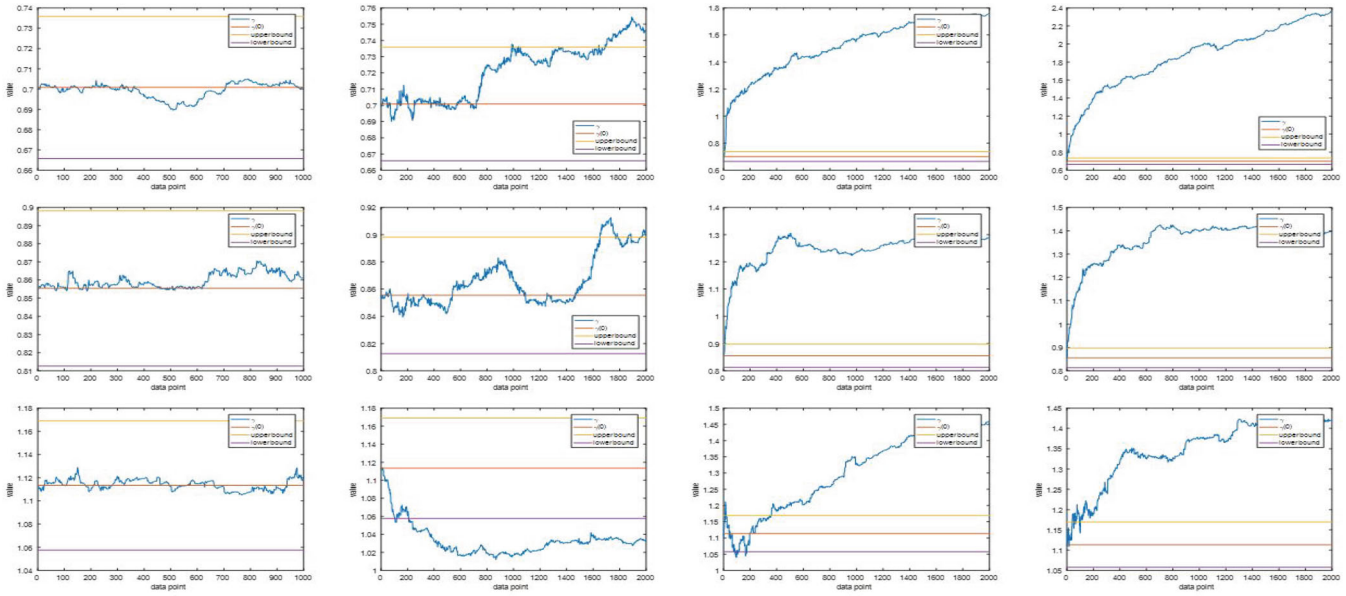

FIGURE 6 Trajectories of $\gamma(k)$: 1st column: nominal case, 2nd–4th column: faulty cases 1–3. 1st row: all RT, 2nd row: c-ss RT, 3rd row: e-ss RT

TABLE 7 Escape time t_e for different damages and techniques

Damage	all PCA	c-ss PCA	e-ss PCA	all RT	c-ss RT	e-ss RT
1	0.88	-	0.91	6.76	6.52	0.82
2	0.04	0.04	0.03	0.05	0.05	1.55
3	0.02	0.01	0.01	0.02	0.02	0.26

Note: Results show that, while for Damages 2 and 3 (the strongest damages) all methods work well without substantial differences, for Damage 1 (the weakest damage and therefore the most difficult to be detected) c-ss PCA fails to detect the damage, all RT and c-ss RT are characterized by a very large escape time, while all PCA, e-ss PCA and e-ss RT are characterized by a similar and small escape time. Since e-ss PCA and e-ss RT use only 5 sensors and all PCA uses 40 sensors, we can state that the e-ss PCA and e-ss RT outperform all other methods in detecting the weak Damage 1.

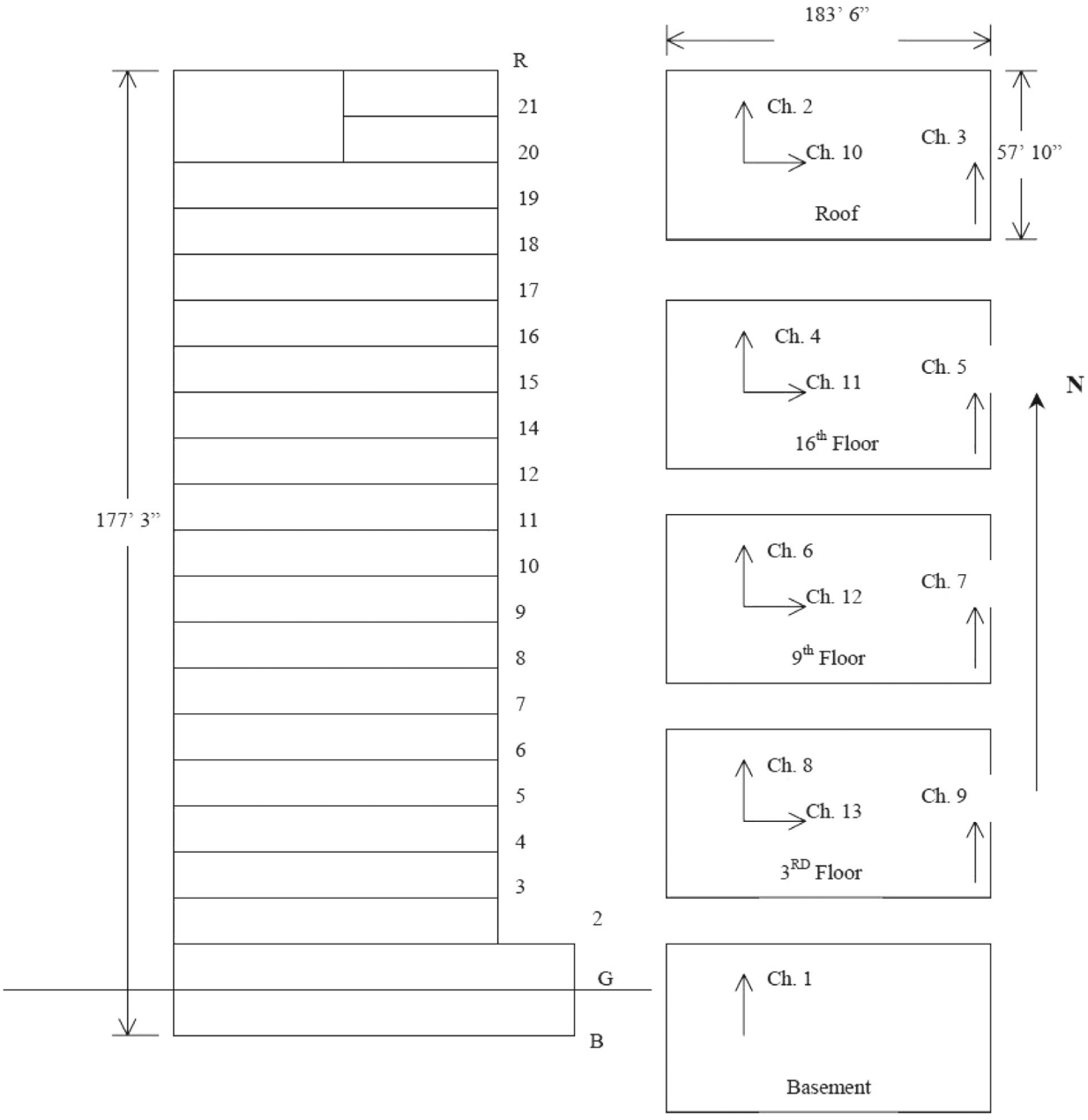

FIGURE 7 Accelerometers placement inside Sheraton hotel

5.3 | Sheraton hotel use-case

The dataset is obtained from LANL teams who measured the vibrational responses on the Sheraton Hotel at Universal City in North Hollywood, California. The hotel is a twenty-one story ductile moment-resisting concrete frame structure that has permanent instrumentation installed on it. There are 13 accelerometers mounted on various locations inside the building, as shown in Figure 7. The accelerometers read the signal caused by background vibration sources such as wind, traffic, and mechanical equipment in the structure, with frequency recorded ranging from 0 to 25 Hz for various amount of times (32, 64, 128, 256, and 512 s). Such measurements generated 10 different datasets, each consisting of 4096 samples. We will consider 2 of the 10 datasets, that is, the ones vibrated with frequency range of 0–6.25 Hz for 256 s, and create a unique dataset of 8192 samples. We will use 6144 samples to train the model, and use the remaining 2048 samples for the validation process. Since in this experimental setup there is no availability of datasets corresponding to a damage, we only validate the predictive model accuracy.

TABLE 8 The results in the table are shown as normalized root mean square error in percentage, that is, (1 - NRMSE) ×100%

all PCA	c-ss PCA	e-ss PCA	opt-PCA	all/e-ss PCA
13.07	84.12*	74.1	88.38	N/A
87.67*	85.18	84.36	88.75	86.85
88.24	89.56*	90*	94.64	88.7
85.76*	62.7	83.35	88.31	N/A
80.49*	76.64	77.41	89.86	79.01
85.48*	84.36	84.41	87.6	N/A
86.83*	87.29*	87.29*	88.6	87.52
82.23	84.79*	84.79*	87.48	79.5
87.18	88.77*	88.77*	89.34	87.94
85.59	88.04*	88.04*	88.13	85.11
83.43	67.86	87.28*	88.17	82.31
86.78	89.72*	89.72*	91.41	88.41
87.67*	87.53*	87.45*	89.25	N/A
80.03	**82.81**	**85.15***	**89.22**	**85.04**

Note: The last row indicates the average prediction accuracy among all 24 components. Results are about sensor selection sets for the PCA-based model. * denotes the best accuracy among all, c-ss and e-ss. Results show that both c-ss and e-ss PCA algorithms always achieve better prediction accuracy with respect to all PCA. The accuracy of e-ss PCA is larger or almost equal to c-ss in almost all cases and in the average. For the prediction of each specific sensor, the optimal sensor selection largely outperforms e-ss and c-ss, probably because e-ss and c-ss assume that the relation among each pair of sensors is linear.

5.3.1 | Predictive model accuracy

We proceed as illustrated above, considering a selection of sets of sensors with cardinality $n^* = 3$ due to the limited number of available accelerometers. Tables 8 and 9 summarize the accuracy of the PCA-based and RT-based predictive models respectively. As above, the last row indicates the average prediction accuracy among all 13 components.

Table 8 shows that both c-ss and e-ss PCA algorithms always achieve better prediction accuracy with respect to all PCA. Also in this case, the accuracy of e-ss PCA is larger or almost equal to c-ss in almost all cases and in the average. For the prediction of each specific sensor, the optimal sensor selection, which is combinatorial and thus impractical when the number of sensors is very large, largely outperforms e-ss and c-ss. Our interpretation is that e-ss and c-ss assume that the relation among each pair of sensors is linear: they fail in the Sheraton hotel dataset because, being it much more complex than the previous structures, the relation among some subsets of sensors is nonlinear.

Interestingly, Table 9 shows that in the RT case all the three algorithms seem to perform very similarly, and outperform PCA-based models (+10% in average accuracy), except for the opt-PCA, although very close. Clearly, in terms of number of sensors, c-ss RT and e-ss RT use less sensors with respect all-RT (3 against 13). Moreover, as discussed above, the use of opt-PCA sensor selection is clearly inapplicable for a large number of sensors. In conclusion, it is interesting that RT-based models perform better than PCA-models when the dataset presents nonlinearities, which can be very well captured by the nonlinear models derived exploiting the RT-based methodology with respect to the linear models obtained via PCA.

5.4 | Nonlinear fault detection via e-ss PCA + PE correction term

In this section, we show the performance of the e-ss PCA + PE algorithm in terms of fault detection performance with respect to both the e-ss PCA algorithm, that is the one that provided the best performance between the proposed ones, and the PLS algorithm, as a comparison with a standard and simple methodology. We show that the e-ss PCA + PE model is more sensitive in detecting faults with respect to the other 2, thus emphasizing that its increase of complexity is justified when compared to the standard PLS, since such simpler algorithm might not give a desirable output.

TABLE 9 The results in the table are shown as normalized root mean square error in percentage, that is, (1 - NRMSE) ×100%

all RT	c-ss RT	e-ss RT	opt-RT	all/ess RT
87.96*	88.12*	88.08*	88.58	N/A
90.01*	88.85	88.22	90.36	89.88
93.22*	93.64*	93.62*	93.83	93.37
88.67*	86.42	85.9	88.92	N/A
88.67*	87.91*	87.85*	89.12	88.65
90.01*	86.22	86.55	89.13	N/A
89.19*	88.59*	88.59*	88.84	89.11
88.74*	86.27	86.27	88.22	87.16
89.87*	89.93*	89.93*	90.44	89.9
86.76*	86.92*	86.92*	87.49	86.88
88.34*	87.07	86.53	88.99	88.45
89.96*	88.95	88.95	90.19	89.55
88.87*	87.85	86.74	89.54	N/A
89.56*	**88.21**	**88.01**	**89.51**	**89.22**

Note: The last row indicates the average prediction accuracy among all 24 components. Results are about sensor selection sets for the PCA-based model. Sensor selection sets are same as Table 8). * denotes the best accuracy among all, c-ss and e-ss. Results show that in the RT case all the three algorithms perform very similarly, and outperform PCA-based models, except for the opt-PCA, although very close. In terms of number of sensors, c-ss RT and e-ss RT use less sensors with respect all-RT (3 against 13). Moreover, the use of opt-PCA sensor selection is clearly inapplicable for a large number of sensors.

To show that e-ss PCA + PE model is more sensitive in detecting faults, we purposely raise the bound $b\%$ to 5%, which is ten times larger than the bound we used in previous cases.

It is worth of notice that our identification technique first identifies the e-ss PCA model and then fits the correction term on the estimation error. Thus, the goal of this section is to show that the addition of the nonlinear correction term increases the performance in detecting faults.

Model predictive accuracy. Before we start with the fault detection procedure, it is important to understand how accurate the models we are going to use are. Table 10 summarizes the comparison of model predictive accuracies between e-ss PCA and e-ss PCA + PE correction terms for all 3 use cases. It can be seen from Table 10 that the e-ss PCA + PE model does usually provides improvements in the model predictive accuracy over the e-ss PCA, although small.

However, we can also see how the e-ss PCA + PE model is able to provide large improvements in the model predictive accuracy w.r.t. the e-ss PCA, see for example, the 8.22% improvement provided on accelerometer 22 in the Irvine Bridge's column and the 11.59% improvement provided on accelerometer 5 in the Sheraton hotel. Given the complexity of these two structures, it is not surprising that the use of the linear model alone is not sufficient to well represent the system. Thus, the nonlinear term introduced by the PE model helps the e-ss PCA model to better match system dynamics.

Regarding the time complexity, the e-ss PCA + PE have additional steps compared to the e-ss PCA, which is the computation of the LM algorithm. However, this calculation converges rapidly, as it is solvable between the time complexity of $\tilde{O}(\varepsilon^{-2})$ or $O(\varepsilon^{-2})$, where ε is the desired model prediction error[61] (in our use case, the algorithm converges at most in 10 iteration steps). Hence, this additional term does not increase significantly the complexity of PCA algorithm, while being able to improve the prediction accuracy.

Fault detection on bookshelf resembling structure use-case. Table 11 shows the escape time (measured in seconds) of the e-ss PCA, the PE and the PLS algorithms for the three damage cases in the bookshelf resembling structure. The sign (-) indicates the incapability of the algorithm to detect faults from the information provided by the corresponding sensor under 5% tolerance bound (i.e., the bound might be too high for the dynamics of the model to pass it).

Notice that the PE correction term provides the best result when compared the e-ss PCA and PLS algorithms, both in producing less number of false positives and faster in detecting faults (characterized by short escape time). The e-ss PCA algorithm generates a lot of false positives, however, it does not mean that the algorithm is not sensitive enough to detect faults: the bound in this simulation is set to be 10 times higher than the one used in previous section. Since the

TABLE 10 The results in the table are shown as normalized root mean square error in percentage, that is, (1 - NRMSE) ×100%

	Bookshelf resembling structure		Irvine Bridge's column		Sheraton hotel	
Sensor n.	**e-ss PCA**	**e-ss PCA +PE**	**e-ss PCA**	**e-ss PCA +PE**	**e-ss PCA**	**e-ss PCA +PE**
1	96.06	96.08	91.23	91.42	74.31	74.38
2	95.32	96.19	85.37	85.48	84.35	84.98
3	93.79	94.18	86.7	92.29*	89.81	91.77*
4	95.79	96.26	91.46	91.57	82.83	82.84
5	94.91	95.79	71.87	72.72	77.5	89.09*
6	95.52	96.36	92.91	93.99*	84.51	84.52
7	94.06	94.36	93.89	93.75	87.11	87.11
8	94.88	96.16*	81.7	85.56*	84.86	84.87
9	96.31	96.28	94.92	94.9	88.82	88.8
10	96.15	96.56	92.76	92.89	88.23	88.29
11	96.32	96.3	83.44	83.78	86.63	86.64
12	91.32	92.68*	90.85	92.03*	89.72	89.74
13	95.8	96.44	92.85	92.57	87.45	87.44
14	95.3	95.97	84.99	86.39*	-	-
15	95.96	96.3	94.85	94.85	-	-
16	92.34	93.08	92.79	92.04	-	-
17	95.75	96.2	86.88	87.49	-	-
18	95.72	96.33	97.13	97.33	-	-
19	95.39	96.08	91.46	92.93*	-	-
20	95.71	96.23	84.93	85.09	-	-
21	96.19	96.23	90.46	92.51*	-	-
22	95.6	96.13	80.3	88.52*	-	-
23	95.92	96.33	87.78	87.59	-	-
24	95	96.32*	88.61	92.74*	-	-
25	-	-	74.73	85.7*	-	-
26	-	-	92.99	94.21*	-	-
27	-	-	79.81	82.4*	-	-
28	-	-	93.41	92.85	-	-
29	-	-	76.91	82.25*	-	-
30	-	-	94.5	95.87*	-	-
31	-	-	89.74	89.62	-	-
32	-	-	92.02	94.04*	-	-
33	-	-	82.94	83.21	-	-
34	-	-	90.73	91.61	-	-
35	-	-	90.45	90.44	-	-
36	-	-	84.81	85.69	-	-
37	-	-	89.18	91.06*	-	-
38	-	-	92.27	91.98	-	-
39	-	-	81.77	81.4	-	-
40	-	-	81.86	82.06	-	-
Average	95.21	95.78	87.95	89.37*	85.08	85.42

Note: Results are about sensor selection sets for the PCA-based and the PCA + PE-based models only with the e-ss over the 3 use cases. * denotes a significant improvement in terms of %. Results show that the e-ss PCA + PE model does usually provides improvements in the model predictive accuracy over the e-ss PCA, although small. However, it provides in some cases large improvements in the model predictive accuracy w.r.t. the e-ss PCA, see for example, the 8.22% improvement provided on accelerometer 22 in the Irvine Bridge's column and the 11.59% improvement provided on accelerometer 5 in the Sheraton hotel.

TABLE 11 Escape time (in seconds) for the e-ss PCA, the PE correction term, the PLS model and the e-ss PCA + PE correction term for the 3 damages in the bookshelf resembling structure

Damage 1				
N.	**e-ss PCA**	**PE**	**e-ss PLS**	**e-ss PCA + PE**
1	2.988	0.820	3.6409	1.3226
2	3.351	1.306	1.8185	-
3	2.048	0.008	5.9815	4.8076
4	1.022	0.542	0.062	0.4168
5	4.096	0.430	4.187	7.996
6	3.351	-	4.143	-
7	2.048	0.006	1.3023	-
8	1.022	0.010	0.3821	1.9018
9	-	0.006	6.2096	1.3206
10	-	6.854	1.2763	-
11	-	0.006	0.03	0.6072
12	-	0.010	-	7.8357
13	-	0.008	1.5404	0.008
14	-	0.008	1.0823	-
15	-	1.504	0.018	1.7956
16	-	0.862	0.016	7.8497
17	4.305	0.470	0.118	0.01
18	4.281	0.386	-	3.7575
19	-	0.006	0.8622	0.002
20	-	0.006	0.144	0.008
21	-	0.010	0.062	0.004
22	-	0.012	0.8242	7.984
23	-	0.008	7.1938	0.008
24	-	0.008	0.5321	0.004
Damage 2				
e-ss PCA	PE	e-ss PLS	e-ss PCA + PE	
3.341	0.008	6.5296	0.008	
4.229	0.008	2.5046	0.004	
2.555	0.006	-	0.008	
0.271	0.302	0.078	5.7936	
6.024	-	0.03	7.998	
4.229	0.008	-	0.004	
2.555	7.694	-	-	
0.271	-	0.6062	1.2265	
-	0.006	1.1563	0.008	
-	0.006	0.3561	-	
-	0.010	0.038	3.8297	
-	0.074	0.03	3.9599	

TABLE 11 Continued

Damage 2			
e-ss PCA	**PE**	**e-ss PLS**	**e-ss PCA + PE**
-	0.008	0.2681	0.004
-	0.008	4.5751	0.008
-	0.008	0.016	0.008
-	0.556	0.012	0.7154
6.415	0.010	0.3021	0.012
-	-	-	1.0501
-	0.008	0.8162	0.004
-	0.008	3.847	0.008
6.653	0.008	0.044	0.008
-	0.432	0.3981	7.8257
7.726	0.006	0.3321	0.004
-	0.010	0.2761	0.4309
Damage 3			
e-ss PCA	PE	e-ss PLS	e-ss PCA + PE
2.429	1.062	0.8282	0.002
2.657	1.224	4.8132	0.004
1.022	0.790	4.4291	0.004
0.110	0.130	2.3806	7.479
2.675	2.697	0.2941	7.998
1.080	0.582	0.9062	0.004
1.022	3.807	6.8897	0.008
0.110	0.418	1.3763	1.521
2.481	1.092	0.4821	0.008
2.677	1.272	1.1343	0.008
4.846	0.008	0.3921	3.8297
3.593	1.274	0.6062	3.9519
2.481	0.006	0.5121	0.008
2.677	1.694	1.7884	0.004
4.846	0.428	0.4501	0.004
3.593	1.234	0.4281	0.7154
3.439	0.704	1.6184	0.012
1.860	1.706	3.809	1.0501
-	0.008	1.6244	0.004
-	0.006	1.8645	0.008
1.589	0.008	0.6242	0.008
1.860	0.008	0.7162	1.426
-	0.008	0.5561	0.004
-	0.008	0.196	0.4329

Note: – denotes a false positive. Results show that the PE correction term provides the best result when compared the e-ss PCA and PLS algorithms, both in producing less number of false positives and faster in detecting faults (characterized by short escape time).

TABLE 12 Escape time for the PE correction term model and e-ss PCA for 3 different damage cases

	e-ss PCA			PE			e-ss PCA + PE		
Sensor n.	**Damage 1**	**Damage 2**	**Damage 3**	**Damage 1**	**Damage 2**	**Damage 3**	**Damage 1**	**Damage 2**	**Damage 3**
1	6.1255	-	0.0157	-	0.0664	0.0274	1.7578	0.1484	0.1172
2	-	0.2784	0.2902	0.5315	0.1368	0.1212	1.4297	4.2656	0.7813
3	0.8863	0.3412	0.251	-	0.1446	0.1759	1.7891	0.1446	0.1172
4	3.9765	0.0275	0.0118	-	0.2775	0.2384	1.6953	0.1563	0.1172
5	0.9255	0.0196	0.0078	-	0.2228	0.1954	1.8594	0.1172	0.125
6	3.4314	0.1412	0.4	-	0.8324	0.6527	1.6953	0.1485	0.1172
7	-	-	0.0118	7.871	0.0899	0.0664	1.6797	0.1563	0.1172
8	2.7373	0.0039	0.0235	-	0.3009	0.3009	1.8594	0.1485	0.125
9	1.3176	0.0157	0.0196	-	0.1485	0.0743	1.6953	0.1563	0.1172
10	-	-	0.0118	7.9961	0.0664	0.0469	1.6875	0.1915	0.1328
11	6.3725	0.0039	0.0118	-	0.254	0.2189	1.8594	0.1915	0.125
12	3.4314	0.1412	0.4	-	0.2736	0.2423	1.7656	0.1172	0.125
13	-	-	0.0039	-	0.1133	0.0821	1.6953	0.1563	0.1328
14	-	-	0.0157	-	2.1612	1.5164	1.9219	0.1563	0.1328
15	1.4627	0.4667	0.3961	-	0.1602	0.1251	1.6875	0.1446	0.1172
16	-	0.0314	0.0118	-	0.1212	0.086	1.6953	0.1563	0.125
17	5.9922	0.8039	0.4	-	0.3596	0.3127	1.6875	0.1563	0.1172
18	0.7451	0.3529	0.7059	-	0.1524	0.1915	1.7891	0.1446	0.1172
19	-	0.0078	0.001	-	0.1212	0.0821	1.7969	0.1641	0.1953
20	5.9922	0.8039	0.4	-	0.0977	0.0899	1.6953	0.1563	0.1172
21	-	0.0196	0.0118	-	0.7777	1.0904	1.7891	0.1446	0.125
22	0.4353	2.902	1.6784	-	0.3244	0.9692	1.8125	0.1563	0.125
23	4.0627	0.0039	0.0118	7.7968	1.0708	0.8676	1.6797	0.1563	0.1172
24	1.8196	2.5255	1.9569	-	1.4617	1.704	1.7734	0.1563	0.125
25	-	-	0.0118	-	0.0703	0.043	1.8828	0.1406	0.0938
26	-	0.0941	0.2471	-	4.85	4.9946	1.6953	0.1563	0.1172
27	4.1529	-	0.0078	-	0.6527	0.3517	1.8516	0.1212	0.125
28	-	0.1451	0.4039	-	1.5476	2.1729	1.6797	0.1563	0.1172
29	1.7059	0.0275	0.0196	-	0.6448	0.5159	1.8516	0.1563	0.125
30	-	0.0941	0.2471	-	0.8481	0.8403	1.7109	0.1172	0.1172
31	-	-	0.0118	-	0.0899	0.0625	1.7578	0.1563	0.1172
32	-	0.0941	0.2471	-	0.5042	0.6175	1.7891	0.1212	0.125
33	2.7961	0.5451	0.4157	-	0.7308	0.555	1.8047	0.1563	0.125
34	-	0.2196	0.4863	-	0.4064	0.512	1.7578	0.1446	0.125
35	1.8627	0.0431	0.0314	7.9805	6.0342	5.1861	1.6875	0.1563	0.1172

TABLE 12 Continued

	e-ss PCA			PE			e-ss PCA + PE		
Sensor n.	**Damage 1**	**Damage 2**	**Damage 3**	**Damage 1**	**Damage 2**	**Damage 3**	**Damage 1**	**Damage 2**	**Damage 3**
36	-	0.0196	0.0118	-	0.1172	0.1133	1.8516	0.1719	0.2188
37	-	-	-	0.3361	1.7509	3.0132	-	-	0.167
38	-	-	0.0039	-	0.1172	0.086	1.6953	0.1563	0.125
39	-	-	6.7373	0.3596	0.3439	0.4963	1.6953	0.1641	0.1328
40	-	0.2784	0.2902	0.0352	0.0743	0.043	1.6797	0.1641	0.1953

Note: – denotes the false positives. Results show that when the damages are strong enough (Damage 2 and Damage 3) it can be seen that the PE-based models are able to detect fault presence in each accelerometer flawlessly w.r.t. the e-ss PCA, and it is done almost immediately when the faulty signal is injected into the system (it can be seen from the short escape time), thus improving the damage detection sensitivity of the e-ss PCA algorithm. When the damage is weak (Damage 1) both the e-ss PCA and the PE (alone) algorithms do suffer from a lot of false positive cases, while their combination allows to detect almost all faults, hence compensating the lacks of the single methodologies.

dynamic growth for the e-ss PCA model is relatively slow than the other two methods, it fails to raise the alarm on a very loose bound. In practical uses, the bounds are defined by the user: tight bounds might raise the alarm faster but more susceptible to the false positives, while loose bounds might decrease the false positives but increase the chance of false negatives. In particular, being able to detect fault faster in a loose bound is an advantage to avoid both false positives and negatives.

Fault detection on UCI Bridge Column use-case. Table 12 shows the escape time for all 40 sensors installed on the bridge's column for the e-ss PCA model, the PE correction term model, and their combination. As in the previous section, the (−) indicates that the algorithm failed to detect a fault on that specific accelerometer.

In this case, we are not able to provide the results for the PLS algorithm as the algorithm generates a lot of false negatives on nominal cases even when the bound is raised to 50%. We claim that this is due to the nature of the data, which is much more complex than the previous case, and hence the dynamics learned by the PLS model behave in an unexpected way.

It is worth of notice that, when the damages are strong enough (i.e., Damage 2 and Damage 3), it can be seen that the PE-based models are able to detect fault presence in each accelerometer flawlessly w.r.t. the e-ss PCA, and it is done almost immediately when the faulty signal is injected into the system (it can be seen from the short escape time). This is an evidence that the PE correction term does improve the damage detection sensitivity of the e-ss PCA algorithm.

However, when the damage is weak (i.e., Damage 1), both the e-ss PCA and the PE (alone) algorithms do suffer from a lot of false positive cases. In particular, the PE model induces more false positive cases than the e-ss PCA model. It is interesting to notice that the PE model is able to detect the presence of faults on several accelerometers which are undetected by the e-ss PCA (see accelerometer numbers 2, 7, 10, 37, 39, and 40), while the integration of the 2 (i.e., the e-ss PCA + PE) compensate the missing of the single methodology.

6 | CONCLUSIONS

In this article we addressed the problem of structural damage detection, and proposed four contributions: (1) a novel damage detection methodology based on the combination of Kalman filtering with a model identification algorithm based on AR system identification, PCA, and RTs; (2) a novel entropy-based sensors selection algorithm; (3) a new nonlinear damage detection methodology that extends the PCA-based one with the addition of a PE term; (4) the validation of our techniques on three experimental datasets.

All experimental setups show that our novel e-ss approach based on PCA models is a valid alternative to classical heuristic c-ss, providing in almost all cases better results in terms of predictive model accuracy, and outperforming c-ss in the average towards the (combinatorial) optimal sensor selection (+50% both in Sections 5.1 and 5.2). The validation in Section 5.3 also shows that when the system presents nonlinearities our novel RT-based approaches outperform

PCA-based models. The damage detection validation in Sections 5.1 and 5.2 shows that both e-ss PCA and e-ss RT perform better than c-ss, improving the prediction accuracy and the damage detection sensitivity while reducing the number of sensors. Finally, the validation in Section 5.4 shows how, with a small increase in the model complexity due to the addition of a PE nonlinear term, the e-ss PCA + PE model improves the fault detection performance on complex structures when nonlinearities become strong.

ACKNOWLEDGMENTS

This work was supported by the Italian Government under Cipe resolution n.135 (December 21, 2012), project *INnovating City Planning through Information and Communication Technologies* (INCIPICT), by the Verification and Validation of Automated Systems' Safety and Security (VALU3S) project under the call H2020-ECSEL-2019-2-RIA, and by the Intelligent Reliability 4.0 (iRel40) project under the call H2020-ECSEL-2019-1-IA. Open Access Funding provided by Universita degli Studi dell'Aquila within the CRUI-CARE Agreement. [Correction added on 18 May 2022, after first online publication: CRUI funding statement has been added.]

CONFLICT OF INTEREST

The authors declare that there is no conflict of interests for this article.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in the Engineering Institute of Los alamos National Laboratory (LANL) repository at https://www.lanl.gov/projects/national-security-education-center/engineering/software/shm-data-sets-and-software.php, Reference number 45.

ORCID


Francesco Smarra https://orcid.org/0000-0002-2715-9447

REFERENCES

1. Tokognon CA, Gao B, Tian GY, Yan Y. Structural health monitoring framework based on Internet of Things: a survey. *IEEE Internet Things J*. 2017;4(3):619-635.
2. Noel AB, Abdaoui A, Elfouly T, Ahmed MH, Badawy A, Shehata MS. Structural health monitoring using wireless sensor networks: a comprehensive survey. *IEEE Commun Surv Tutor*. 2017;19(3):1403-1423.
3. Fasel TR, Sohn H, Farrar CR. Application of frequency domain ARX models and extreme value statistics to damage detection. *Smart Struct Mater 2003 Smart Syst Nondestruct Eval Civil Infrastruct*. 2003;5057:145-156.
4. Balsamo L, Betti R. Data-based structural health monitoring using small training data sets. *Struct Control Health Monit*. 2015;22(10):1240-1264.
5. Bodeux JB, Golinval JC. Application of ARMAV models to the identification and damage detection of mechanical and civil engineering structures. *Smart Mater Struct*. 2001;10(3):479.
6. Zhao R, Yan R, Chen Z, Mao K, Wang P, Gao RX. Deep learning and its applications to machine health monitoring. *Mech Syst Signal Process*. 2019;115:213-237.
7. Tibaduiza Burgos DA, Gomez Vargas RC, Pedraza C, Agis D, Pozo F. Damage identification in structural health monitoring: a brief review from its implementation to the use of data-driven applications. *Sensors*. 2020;20(3):733.
8. Neves AC, Gonzalez I, Leander J, Karoumi R. Structural health monitoring of bridges: a model-free ANN-based approach to damage detection. *J Civ Struct Heal Monit*. 2017;7(5):689-702.
9. Azimi M, Pekcan G. Structural health monitoring using extremely compressed data through deep learning. *Comput Aided Civ Inf Eng*. 2020;35(6):597-614.
10. Tang Z, Chen Z, Bao Y, Li H. Convolutional neural network-based data anomaly detection method using multiple information for structural health monitoring. *Struct Control Health Monit*. 2019;26(1):e2296.
11. Chang CM, Lin TK, Chang CW. Applications of neural network models for structural health monitoring based on derived modal properties. *Measurement*. 2018;129:457-470.
12. Hastie T, Tibshirani R, Friedman J, Hastie T, Friedman J, Tibshirani R. *The Elements of Statistical Learning*. Vol 2. Springer; 2009.
13. Isermann R. *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Springer Science & Business Media; 2006.
14. Deng N, Wang JG, Szostak-Chrzanowski A. *Dam Deformation Analysis Using the Partial Least Squares Method*. Citeseer; 2008.
15. Jackson JE. *A User's Guide to Principal Components*. John Wiley & Sons; 2005.
16. Krishnan M, Bhowmik B, Tiwari AK, Hazra B. Online damage detection using recursive principal component analysis and recursive condition indicators. *Smart Mater Struct*. 2017;26(8):085017.
17. Lakshmi K, Rama Mohan Rao A. A robust damage-detection technique with environmental variability combining time-series models with principal components. *Nondestruct Test Eval*. 2014;29(4):357-376.

18. Dunia R, Qin SJ, Edgar TF, McAvoy TJ. Identification of faulty sensors using principal component analysis. *AICHE J*. 1996;42(10):2797-2812.
19. Nguyen VH, Golinval JC. Fault detection based on kernel principal component analysis. *Eng Struct*. 2010;32(11):3683-3691.
20. Kesavan KN, Kiremidjian AS. A wavelet-based damage diagnosis algorithm using principal component analysis. *Struct Control Health Monit*. 2012;19(8):672-685.
21. Pan VY, Chen ZQ. The complexity of the matrix eigenproblem. Proceedings of the 31st Annual ACM Symposium on Theory of Computing; 1999:507-516.
22. Lee YK, Lee ER, Park BU. Principal component analysis in very high-dimensional spaces. *Stat Sin*. 2012;22(3):933-956.
23. Johnstone IM, Paul D. Pca in high dimensions: an orientation. *Proc IEEE*. 2018;106(8):1277-1292.
24. Shamaiah M, Banerjee S, Vikalo H. Greedy sensor selection under channel uncertainty; 2012:2572-2577; IEEE.
25. Gupta V, Chung TH, Hassibi B, Murray RM. On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage. *Automatica*. 2006;1(4):376-379.
26. Liu S, Chepuri SP, Fardad M, Maşazade E, Leus G, Varshney PK. Sensor selection for estimation with correlated measurement noise. *IEEE Trans Signal Process*. 2016;64(13):3509-3522.
27. Papadimitriou C, Beck JL, Au SK. Entropy-based optimal sensor location for structural model updating. *J Vib Control*. 2000;6(5):781-800.
28. Yuen KV, Katafygiotis LS, Papadimitriou C, Mickleborough NC. Optimal sensor placement methodology for identification with unmeasured excitation. *J Dyn Syst Meas Control*. 2001;123(4):677-686.
29. Papadimitriou C. Optimal sensor placement methodology for parametric identification of structural systems. *J Sound Vib*. 2004;278(4):923-947. doi:10.1016/j.jsv.2003.10.063
30. Yuen KV, Kuok SC. Efficient Bayesian sensor placement algorithm for structural identification: a general approach for multi-type sensory systems. *Earthq Eng Struct Dyn*. 2015;44(5):757-774.
31. Zhang J, Maes K, De Roeck G, Reynders E, Papadimitriou C, Lombaert G. Optimal sensor placement for multi-setup modal analysis of structures. *J Sound Vib*. 2017;401:214-232.
32. Lin JF, Xu YL, Law SS. Structural damage detection-oriented multi-type sensor placement with multi-objective optimization. *J Sound Vib*. 2018;422:568-589.
33. Gomes GF, de Almeida FA, de Silva Lopes Alexandrino P, de Cunha SS Jr, de Sousa BS, Ancelotti AC Jr. A multiobjective sensor placement optimization for SHM systems considering Fisher information matrix and mode shape interpolation. *Eng Comput*. 2019;35:519-535.
34. Smarra F, Jain A, De Rubeis T, Ambrosini D, D'Innocenzo A, Mangharam R. Data-driven model predictive control using random forests for building energy optimization and climate control. *Appl Energy*. 2018;226:1252-1272.
35. Smarra F, Di Girolamo GD, De Iuliis V, Jain A, Mangharam R, D'Innocenzo A. Data-driven switching modeling for MPC using regression trees and random forests. *Nonlinear Anal Hybrid Syst*. 2020;36:100882.
36. Di Girolamo GD, Smarra F, Gattulli V, Potenza F, Graziosi F, D'Innocenzo A. Data-driven optimal predictive control of seismic induced vibrations in frame structures. *Struct Control Health Monit*. 2020;27(4):1-23.
37. Imbimbo BP, Imbimbo E, Daniotti S, Verotta D, Bassotti G. A new criterion for selection of pharmacokinetic multiexponential equations. *J Pharm Sci*. 1988;77(9):784-789.
38. Moreira AB, Oliveira HP, Atvars TD, et al. Direct determination of paracetamol in powdered pharmaceutical samples by fluorescence spectroscopy. *Anal Chim Acta*. 2005;539(1-2):257-261.
39. De Paepe P, Van Hoey G, Belpaire FM, Rosseel MT, Boon PA, Buylaert WA. Relationship between etomidate plasma concentration and EEG effect in the rat. *Pharm Res*. 1999;16(6):924-929.
40. Kim T, San Kim D, Kwon J, Lee H. Degenerate polyexponential functions and type 2 degenerate poly-Bernoulli numbers and polynomials. *Adv Differ Equ*. 2020;2020(1):1-12.
41. Mager DE, Jusko WJ. General pharmacokinetic model for drugs exhibiting target-mediated drug disposition. *J Pharmacokinet Pharmacodyn*. 2001;28(6):507-532.
42. Giurgica-Tiron T, Hindy Y, LaRose R, Mari A, Zeng WJ. Digital zero noise extrapolation for quantum error mitigation; 2020:306-316; IEEE.
43. Farrar CR, Nix DA, Duffey TA, Cornwell PJ, Pardoen GC. Damage identification with linear discriminant operators; 1999.
44. Tjen J, Smarra F, D'Innocenzo A. An entropy-based sensor selection algorithm for structural damage detection; 2020:1566-1571; IEEE.
45. Engineering Institute of Los Alamos National Laboratory (LANL). Accessed April 24, 2018. https://www.lanl.gov/projects/national-security-education-center/engineering/software/shm-data-sets-and-software.php
46. Smith LI. A tutorial on principal components analysis. Technical report; 2002.
47. Jolliffe IT. *Principal Component Analysis*. Springer; 2002.
48. Doraiswami R, Stevenson M, Diduch C. *Identification of physical systems: Applications to Condition Monitoring, Fault Diagnosis, Soft Sensor and Controller Design*. John Wiley & Sons; 2014.
49. Gelb A. *Applied Optimal Estimation*. MIT Press; 1974.
50. Simon D. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. John Wiley & Sons; 2006.
51. Breiman L. *Classification and Regression Trees*. Routledge; 2017.
52. Willsky AS. A survey of design methods for failure detection in dynamic systems. *Automatica*. 1976;12:601-611.
53. Hall MA. *Correlation-Based Feature Selection for Machine Learning*. PhD thesis; 1999.
54. Haindl M, Somol P, Ververidis D, Kotropoulos C. Feature selection based on mutual correlation; 2006:569-577; Springer.
55. Eshima N, Tabata M. Entropy coefficient of determination for generalized linear models. *Comput Stat Data Anal*. 2010;54(5):1381-1389.

56. Rubinstein RY, Kroese DP. The cross-entropy method: a unified approach to combinatorial optimization. *Monte-Carlo Simulation, and Machine Learning*. Vol 133. Springer; 2004.
57. Asmussen S, Kroese DP, Rubinstein RY. Heavy tails, importance sampling and cross–Entropy. *Stoch Model*. 2005;21(1):57-76.
58. Levenberg K. A method for the solution of certain non-linear problems in least squares. *Q Appl Math*. 1944;2(2):164-168.
59. Moré JJ. The Levenberg-Marquardt algorithm: implementation and theory; 1978:10-116; Springer.
60. Sipser M. *Introduction to the Theory of Computation*. 1st ed. International Thomson Publishing; 1996.
61. Yamashita N, Fukushima M. On the rate of convergence of the Levenberg-Marquardt method. *Computing*. 2011;15:239-249.

How to cite this article: Smarra F, Tjen J, D'Innocenzo A. Learning methods for structural damage detection via entropy-based sensors selection. *Int J Robust Nonlinear Control*. 2022;32(10):6035-6067. doi: 10.1002/rnc.6124