



ORIGINAL PAPER

Pavle Milicevic<sup>ID</sup> · Okay Altay<sup>ID</sup>

# Data generation framework for inverse modeling of nonlinear systems in structural dynamics applications

Received: 31 October 2022 / Revised: 31 January 2023 / Accepted: 25 February 2023  
© The Author(s) 2023

**Abstract** In structural dynamics, response modeling relies on parameters, which are to be identified by experiments. However, for satisfactory results, the design of such experiments is laborious and requires a comprehensive physical insight, which is limited. Furthermore, accurate models are high dimensional and can operate only with a large set of parameters, which increases the experimental effort even more. Efficient data sampling methods have been addressed in studies within areas of design of experiments and active learning. However, generating a data set for nonlinear dynamic systems poses an increased degree of difficulty, since the system needs to be guided through unknown dynamics to collect the desired data points. In this paper, we address this challenge by introducing a theoretical data generation framework for testing-integrated modeling. In the proposed framework, we use feedforward neural networks (FNNs) for inverse modeling of the nonlinear restoring force of the systems. By sequentially evaluating the accuracy of the trained model on a given test data set, the excitation signal applied on the system is adapted to generate optimal response data which allow the FNN model to learn the restoring force behavior. Hence, data generation is posed as an optimization problem and pattern search algorithm is used for sampling. The performance of the proposed framework is evaluated, and it is shown that it outperforms unsupervised sampling methods.

## 1 Introduction

In structural dynamics, experiments are necessary in order to determine the parameters of nonlinear systems, such as for the optimal design of dampers [1] or within the scope of vibration-based identification [2–4]. A significant effort has been spent on the development of experimental setups in civil engineering, such as using large-scale actuators and shaking tables. With real-time hybrid simulation, these experiments can also be combined with numerical models, such as for the investigation of soil–structure interaction [5]. However, the existing physical testing procedures are limited to pure performance assessment and do not result in obtaining a model of the tested specimen.

Machine learning methods are getting increasingly popular for modeling of nonlinear dynamical systems and are implemented in various applications, such as anti-seismic materials and devices, cf. [6] and the references therein. These models are suitable for fast and accurate modeling. However, a common bottleneck is the required amount and quality of data, which are necessary for the training of the machine learner. Specifically in structural dynamics applications, generation of real response data is hard and expensive to obtain which motivates developments of efficient data generation methods.

---

P. Milicevic · O. Altay (✉)  
Department of Civil Engineering, RWTH Aachen University, Mies-van-der-Rohe-Str. 1, 52074 Aachen, Germany  
E-mail: altay@lbb.rwth-aachen.de

P. Milicevic  
E-mail: milicevic@lbb.rwth-aachen.de

The need for more efficient data collection has been addressed by *design of experiments* (DoE) where *unsupervised* methods, such as random and Latin hypercube sampling (LHS) and sequential space-filling methods (e.g., Sobol sequence), are used [7]. *Active learning* or adaptive sampling aims to further improve the data collection by considering the current performance of the model, based around some measure of informativeness, in order to gather data [8]. There are numerous active learning strategies which can be employed based on the application [9]. These methods are being researched for modern machine learning methods, such as the physics-informed neural networks (PINNs) for forward and inverse modeling. For instance, Nabian et al. proposed importance sampling to improve the training performance of PINNs [10]. Further methods, including non-adaptive and residual-based adaptive sampling, have been studied by Wu et al. [11]. While successful in their intended application, these sampling methods are developed for static mappings where the desired states can be directly collected, and the function evaluations are inexpensive. Hence, they are not for sampling of dynamic data and for applications where the cost of data acquisition is a limiting factor, as it is the case in this paper.

Another approach in active learning strategies dealing with identification and modeling of nonlinear dynamic systems is using uncertainty, which is a convenient byproduct of probabilistic machine learning methods, such as Gaussian Process (GP). For instance, Zhao et al. used active learning approach based on GP regression to reduce the number of dissipative particle dynamics simulations for multiscale modeling of non-Newtonian fluids [12]. Despite its successful applications in active learning, GP is known for not scaling well with the amount of provided data. Similar to GP uncertainty, Belz et al. exploit special properties of local model networks (LMNs), namely local model errors, on which they base the active learning strategy and sample only the points which lay in the areas with high local errors [13]. However, in our proposed framework, we assume no special properties coming from the model, such as local errors or uncertainty, are available.

In dynamic sampling problems, the system needs to be guided through unknown dynamics in order to explore the desired states. This is attempted by designing excitation signals, which drive the system to cover most of its operating range such as modulated chirp signal [14] and binary signals, such as pseudorandom binary sequences (PRBS) for linear systems and amplitude modulated pseudorandom binary sequences (APRBS) for nonlinear systems [15]. Extensions of these approaches belong to sequential signal design [16] where the previously measured data are taken into account in order to boost diversity of the new data to be collected. However, these approaches belong to the batch methods and do not consider the current performance of the model. Hametner et al. addressed the identification of both static and dynamic nonlinear systems using the Fisher information matrix as an informativeness criteria [17]. However, the Fisher information matrix requires an *a priori* chosen model structure and its linearization, which are both problematic for nonlinear systems as pointed out by Nelles [7]. Furthermore, data generation in dynamical systems has also been addressed in control applications by Buisson-Fenet et al. who proposed several algorithms for active learning using GPs [18]. While the authors work provides a significant improvement over batch data generation methods, as mentioned above, our aim is to develop a framework which assumes that uncertainty information coming from the model is not available.

In some civil engineering applications, such as *structural health monitoring* (SHM), the stream of data is available and needs to be selected efficiently for labeling and interpretation. Active learning strategies have been applied in SHM applications [19–21] and for digital twins [22]. However, the stream of data is not available for the considered problem setup of the present paper and our task is purposeful generation of data.

Despite the aforementioned research efforts and to the best of authors' knowledge, a systematic data generation approach, such as testing-integrated modeling, has not been attempted in the domain of structural dynamics. Aiming to fill this gap, we introduce a theoretical framework for the adaptive data generation to allow automatic modeling of nonlinear dynamical systems, such as structural vibration control devices. The focus of the study is on systems where the specimen mass cannot be decoupled from the restoring force, such as tuned mass dampers (TMDs), hence requiring sampling of dynamic data. Optimized data sampling is coupled with modeling of the nonlinear restoring force using feedforward neural networks (FNNs) and carried out by the pattern search (PS) method. Numerical simulations are conducted to validate the proposed framework.

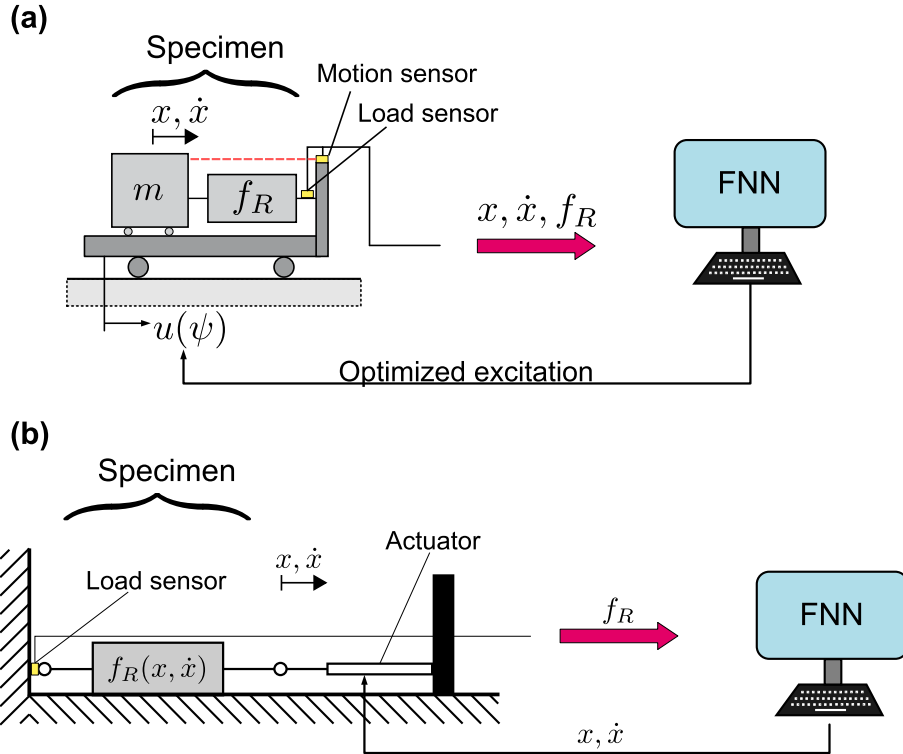
The paper is structured as follows. The modeling setup and the proposed data generation framework are introduced in Sect. 2. The numerical simulations and the comparison to other sampling methods are presented in Sect. 3 on a Duffing oscillator. An engineering application example is presented on a two-story shear frame with a nonlinear TMD. Finally, the contributions of this work are summarized in Sect. 4.

## 2 Methods

### 2.1 Problem setup

The proposed framework is focused on modeling of nonlinear restoring force of oscillating systems, such as damping devices or substructures, using force-state mapping. The advantages of using a force-state mapping are that it fully describes the system dynamics. Furthermore, if the system has a local nonlinearity or is a nonlinear substructure within a larger structure, the modeled force-state behavior can be readily integrated in a numerical simulation [23]. As shown in Fig. 1a, in this work the force-state map is approximated by an FNN, where the data are assumed to be obtained through shaking table testing of the specimen. Here, the specimen motion and the corresponding restoring force are captured by sensors, cf. Sect. 2.4.1. The FNN evaluates its accuracy on an independent data set, and requests, if necessary, more data in order to improve its modeling accuracy. The detailed steps are presented in Sect. 2.3.

It is important to note that the proposed framework also holds for other testing setups, such as damping testing systems or dynamic testing systems for bearings and isolators, where the restoring force is directly identified by applying the velocity and displacement on the system as shown in Fig. 1b. The key difference between the setups in Fig. 1a, b is that the former one contains an oscillating mass, while the latter one does not. The restoring force of the oscillating system, as represented by the specimen shown in Fig. 1a, must be identified by applying an excitation signal and collecting the data for the desired states. However, the system must be guided to the desired states through *unknown dynamics*. On the other hand, the information about the restoring force of the specimen in Fig. 1b can be directly collected for the given pair of velocity and displacement. This key difference requires the dynamic sampling of data for the first setup and motivates our approach. Therefore, in the remaining paper, we focus on the more challenging first setup.



**Fig. 1** **a** Setup 1: A nonlinear oscillator (e.g., a tuned mass damper) is mounted on a shaking table. The restoring force-state mapping is learned by a feedforward neural network (FNN), which automatically controls the excitation signal to generate response data required for the optimum model training; **b** Setup 2: A nonlinear specimen (e.g., a dissipator) is attached to an actuator. Similar to Setup 1, the restoring force-state mapping is learned by an FNN

## 2.2 Mathematical formulation

The equation of motion of the oscillator (Setup 1, cf. Fig. 1a) with unknown components of the restoring force can be written as

$$m\ddot{\mathbf{x}} + \mathbf{f}_R(\dot{\mathbf{x}}, \mathbf{x}) = \mathbf{u}(\boldsymbol{\psi}), \quad (1)$$

where the mass  $m$  is assumed to be known;  $\mathbf{x} \in \mathbb{R}^{N \times 1}$  is the displacement vector recorded at  $N$  time steps; a dot denotes the derivative with respect to time; and  $\mathbf{f}_R \in \mathbb{R}^{N \times 1}$  is the restoring force and includes both linear and nonlinear parts. The applied excitation signal  $\mathbf{u} \in \mathbb{R}^{N \times 1}$  is characterized by its parameters  $\boldsymbol{\psi} \in \mathbb{R}^M$  where  $M$  is the parameter number and represents the sampling space dimension in the proposed framework.

The restoring force  $\mathbf{f}_R$  is modeled with an FNN denoted as  $\mathcal{M}(\dot{\mathbf{x}}, \mathbf{x}, \boldsymbol{\theta})$  where  $\boldsymbol{\theta}$  is the matrix containing model parameters. As shown in Fig. 1a, the model is trained and assessed with sensor data, which are to be collected by experiments on the specimen. Both the training data  $\mathcal{D}_i\{\dot{\mathbf{x}}_i, \mathbf{x}_i, \mathbf{f}_{R,i}\}$  and the test data  $\mathcal{D}_T\{\dot{\mathbf{x}}_T, \mathbf{x}_T, \mathbf{f}_{R,T}\}$  consist of measured velocity and displacement responses with the corresponding restoring force. For the training of the FNN model, the setup can produce various response data sets by updating the excitation signal parameters  $\boldsymbol{\psi}_i$ . The details are presented in Sect. 2.3, and the goals are twofold:

1. To find a training data set  $\mathcal{D}_i$  such that when the model is trained with  $\mathcal{D}_i$  and evaluated on an independent test data set  $\mathcal{D}_T$ , it meets a modeling accuracy with an error tolerance of  $\epsilon$ . We refer to this training data set as the *optimum training data set* and denote it as  $\mathcal{D}_{\text{opt}}$ .
2. To minimize the number of experiments  $i$  needed to find the optimum training data set  $\mathcal{D}_{\text{opt}}$ .

## 2.3 Adaptive data generation framework

In order to find the optimum training data set  $\mathcal{D}_{\text{opt}}$  with the minimum number of tests  $i$ , the framework proposed in this work essentially turns the data generation into an optimization problem. We refer to the model error achieved on the test data set as *adaptation cost function*  $\mathcal{L}_T$ , where  $\mathcal{L}_T : \mathbb{R}^M \rightarrow \mathbb{R}$ . A single evaluation of  $\mathcal{L}_T$  corresponds to the model being evaluated after one experiment and its training with the generated data. The computed value of the function is referred to as *test error*. To achieve the above defined goals, excitation signal parameters  $\boldsymbol{\psi}_i$  are updated sequentially in the direction of minimizing the adaptation cost function  $\mathcal{L}_T$  until the error tolerance  $\epsilon$  is reached. Note that training set  $\mathcal{D}_i$  is not cumulative, meaning that the model  $\mathcal{M}(\dot{\mathbf{x}}, \mathbf{x}, \boldsymbol{\theta})$  is always trained with a single training set  $\mathcal{D}_i$  with constant size and produced with different excitation signal parameters  $\boldsymbol{\psi}_i$ . Hence, the adaptation cost function can be viewed as a measure of information carried by a training data set. The proposed framework is illustrated in Fig. 2 and the detailed description is as follows:

**Step 1:** Choose the type of excitation signal to be used to obtain the training data set and randomly select its initial parameters  $\boldsymbol{\psi}_i$  for  $i = 1$ .

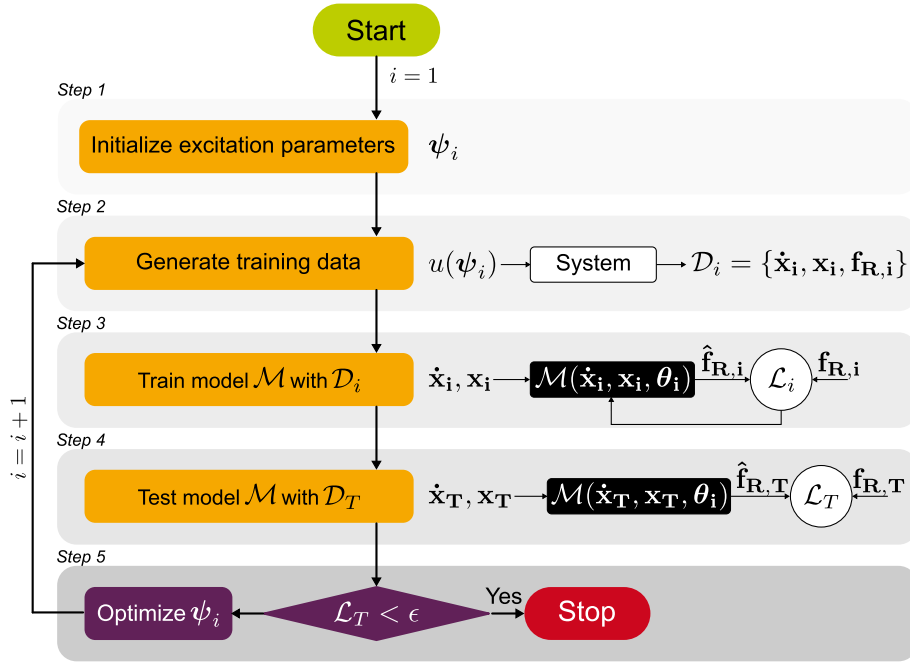
**Step 2:** Excite the specimen with the excitation signal  $\mathbf{u}(\boldsymbol{\psi}_i)$  and obtain the training data set  $\mathcal{D}_i = \{\dot{\mathbf{x}}_i, \mathbf{x}_i, \mathbf{f}_{R,i}\}$  from the recorded response and restoring force.

**Step 3:** Split the obtained training set into training and validation sets. Use velocity and displacement vectors as inputs to model  $\mathcal{M}$  whose parameters are set to initial values  $\boldsymbol{\theta}_0$ . Train the model in a supervised mode by comparing model output vector  $\hat{\mathbf{f}}_{R,i}$  to the true values of the recorded restoring force  $\mathbf{f}_{R,i}$  to obtain the trained model  $\mathcal{M}(\dot{\mathbf{x}}_i, \mathbf{x}_i, \boldsymbol{\theta}_i)$ .

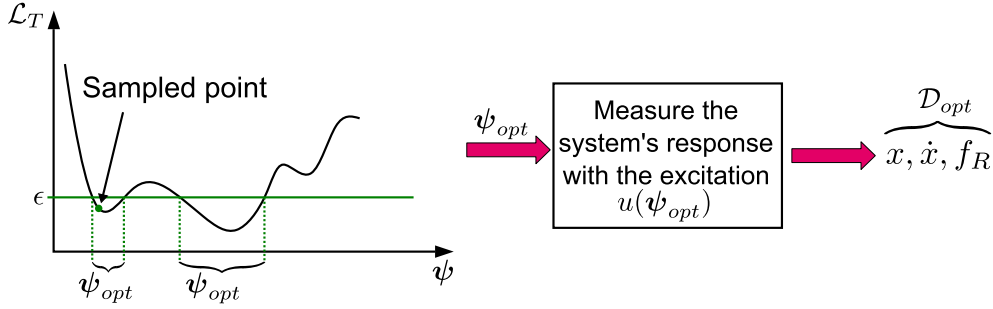
**Step 4:** Test the model on the provided test set  $\mathcal{D}_T = \{\dot{\mathbf{x}}_T, \mathbf{x}_T, \mathbf{f}_{R,T}\}$  and obtain the adaptation cost function  $\mathcal{L}_T$ .

**Step 5:** If  $\mathcal{L}_T$  meets the given tolerance  $\epsilon$ , then stop. Otherwise, optimize the excitation signal parameters  $\boldsymbol{\psi}_i$  and go to **Step 2**. Continue until the tolerance or predefined maximum number of experiments  $i$  is reached.

It is important to note that the solution found by the proposed framework does not necessarily correspond to the global minimum of the adaptation cost function. As depicted in Fig. 3,  $\mathcal{D}_{\text{opt}}$  is not a single best training set, but each training set produced with the excitation signal parameters  $\boldsymbol{\psi}_{\text{opt}}$  such that  $\mathcal{L}_T(\boldsymbol{\psi}_{\text{opt}}) < \epsilon$ . Furthermore, for the sake of clarity, it should be emphasized that the test set is to be provided by the user prior to the execution of the proposed framework and obtained, for instance, from previous performance assessment tests of the specimen corresponding to its operational conditions and existing standards. Hence, it is not generated in the Step 4 of the procedure, but only used to evaluate the model accuracy.



**Fig. 2** Workflow chart of the of the proposed adaptive data generation framework for training of FNNs to model nonlinear dynamical systems



**Fig. 3** Optimum training data set  $\mathcal{D}_{opt}$  produced by an excitation signal with optimum parameters  $\psi_{opt}$  such that the adaptation cost function is smaller than the error tolerance  $\mathcal{L}_T(\psi_{opt}) < \epsilon$

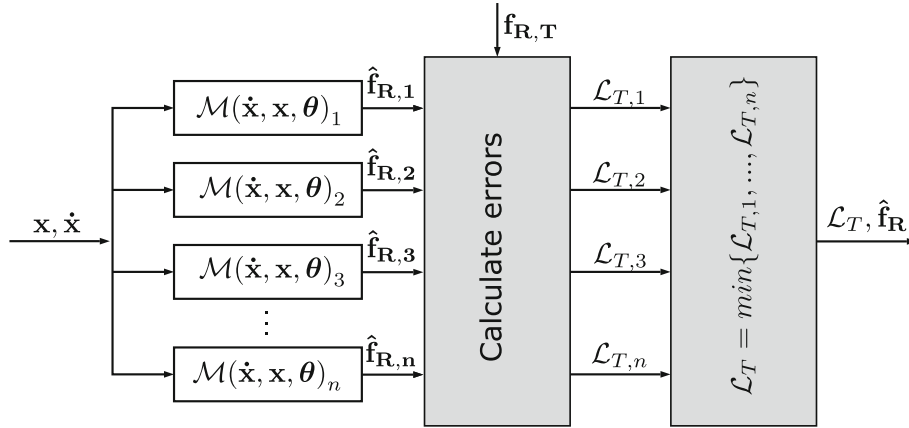
## 2.4 Feedforward neural network

For modeling, in this work, we consider FNNs as data-driven methods which are a type of neural networks with forward information propagation through layers, meaning no information is looped back or delayed. FNNs are particularly suitable for force-state mapping and widely used in nonlinear dynamics applications for system identification and control [24]. They have also shown to be well resistant to noise [7], which can even help make their training more robust and prevent overfitting [24]. For the readers, who are not familiar with machine learning, the authors recommend general literature, such as Goodfellow et al. [25]. The structure of the neural network is commonly referred to as an architecture and is denoted in matrix form as

$$\mathbf{a}^l = f(\mathbf{W}^l \mathbf{a}^{l-1} + \mathbf{b}^l), \quad (2)$$

where  $\mathbf{a}^l$  is the output vector of the  $l$ th layer;  $f(\cdot)$  is the activation function of the  $l$ th layer;  $\mathbf{W}^l$  is the weight matrix of the connection between the  $l$ th and the  $(l-1)$ th layers; and  $\mathbf{b}^l$  is the bias vector of the  $l$ th layer.

FNNs are trained using algorithms, which in a supervised training mode compare the network output with the true output and use this error information to update the network parameters (i.e., its weights and biases) in order to minimize the error. It is worth noting that the FNN architectures where linear activation functions are used in the output layer and sigmoid or tangent sigmoid activation functions are used for other layers



**Fig. 4** FNN model ensemble for obtaining the model with the most successful training; by using a model ensemble, information about the quality of the training data set has been made more reliable

are also referred to as multilayer perceptron (MLP) networks and their applications in system modeling and identification can be found in literature, such as [26].

An important issue in the proposed approach is that the model error depends not only on the quality of data. The test error can be decomposed into the *bias* and the *variance* errors where the bias error comes from the mismatch of the model and the true physics, and the variance error comes from the success of the training algorithm and provided data. That means that two identical model architectures and training data sets can lead to different test errors. Hence, training must be repeated several times in order to get the best results. In this paper, our goal is to improve the sampling of the data and not the FNN training convergence. Therefore, we assume that the residual error is purely due to data quality. In other words, by improving only the data set, we try to get the desired model performance. For this assumption to hold true, we need to repeat the training of the model several times and record the best outcome. As shown in Fig. 4, we form a *neural network ensemble*, where several FNNs are trained in parallel with the same data set, and the one with the best modeling performance is used to evaluate the quality of the data set. It should be mentioned that there will always be some remaining bias error, but we expect its effect will be negligible. It is also worth noting that we use strictly force-state mapping and therefore FNNs. However, other configurations are possible where the response of the system being modeled without directly measuring the displacement and the restoring force, such as from excitation signal to displacement, velocity and acceleration responses [27].

#### 2.4.1 Training and test data

To generate the data, the system is first excited by some excitation signal  $\mathbf{u}(\psi)$  and the velocity and displacement responses as well as the restoring responses are then collected by sensors. The displacements can be measured with laser sensors, and velocity can be derived from the displacements. The restoring force measurement can be done with a load cell, such as in [28], or with strain gauges, which are calibrated to determine the shear force at the connection point with the shaking table [5]. As mentioned before, the operating range of the system is assumed to be known, such as from initial performance assessment tests, and the experiments are conducted in this range.

The data set for model testing is generated by a user provided excitation signal corresponding to the operating range of the system. In existing experimental performance assessment procedures, such excitation signals are already being used, e.g., earthquake time histories for testing control devices. These data are independent of the training data sets and allow performance validation of the training set. To ensure the independent test set, a different type of excitation signal can be used to generate the system response than the one used for obtaining the training sets. It is also possible to use the same excitation signal, but then it should be ensured that the two sets are independent, such as by calculating the Euclidean distance between the data points.

### 2.4.2 Pattern search

Pattern search (PS) methods are a class of direct search optimization methods which are distinguished by not explicitly requiring calculation of derivatives. The equations presented in this section are adapted by the authors for the proposed data generation framework. The theoretical background of the equations can be found in the literature, such as on generalized pattern search (GPS) by Torczon [29] and generative set search (GSS) methods by Kolda et al. [30]. We describe the adaptation required for the proposed framework with the basic PS formulation to aid the understanding of the results. This class of methods can be applied to functions where smoothness or even continuity cannot be assumed. In the present optimization problem, the function to be minimized is stochastic both due to the noisy measurement data as well as the choice of the training data and its success in the training of the FNN model. Therefore, a derivative-free optimization method must be used. Population-based optimization methods, such as Genetic Algorithm or Particle Swarm Optimization, would likely lead to better results. However, they require more function evaluations, and therefore, in our problem setup, they would lead to more experiments. In contrast with population-based methods, PS algorithms have proved to be efficient in applications where function evaluations are expensive [31].

In the optimization problem posed by our framework, the adaptive cost function  $\mathcal{L}_T(\boldsymbol{\psi})$  is to be minimized, where  $\boldsymbol{\psi}$  represents a candidate solution and has dimension  $M$  corresponding to the number of design variables. At each iteration, PS generates a finite number of exploratory vectors  $\boldsymbol{\psi}^e$ . This set of exploratory vectors is referred to as a *mesh* and can be expressed as

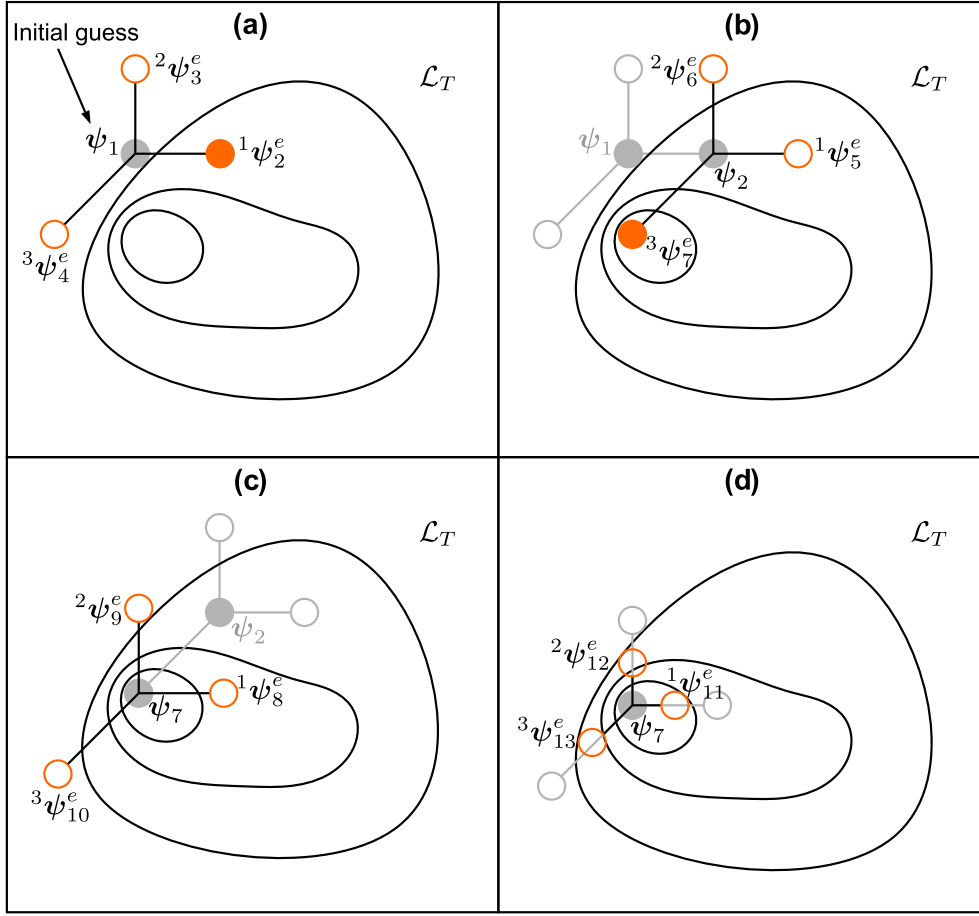
$${}^k\boldsymbol{\psi}_i^e = \boldsymbol{\psi}_i + \Delta \cdot {}^k\mathbf{v}, \quad (3)$$

where  ${}^k\mathbf{v}$  is a set of  $k$  vectors used by the PS algorithm to determine which points should be evaluated and referred to as a *pattern*;  $\Delta$  is the step length and referred to as the *mesh size*. From Eq. (3), it can be seen that the mesh is centered at  $\boldsymbol{\psi}$ . It is defined by the number of independent design variables  $M$  and the positive basis set. So, for example, for a problem with a minimal basis and mesh size  $\Delta = 1$ , mesh will consist of  $M + 1$  vectors and look like as follows:

$${}^1\boldsymbol{\psi}_{i+1}^e = \boldsymbol{\psi}_i + 1 \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad {}^2\boldsymbol{\psi}_{i+2}^e = \boldsymbol{\psi}_i + 1 \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad {}^3\boldsymbol{\psi}_{i+3}^e = \boldsymbol{\psi}_i + 1 \cdot \begin{bmatrix} -1 \\ -1 \end{bmatrix}. \quad (4)$$

If one of the trial points leads to an improved solution such that  $\mathcal{L}_T({}^k\boldsymbol{\psi}_{i+k}^e) < \mathcal{L}_T(\boldsymbol{\psi}_i)$ , then the mesh center is moved such that  $\boldsymbol{\psi}_{i+k} = {}^k\boldsymbol{\psi}_{i+k}^e$ . The mesh is then scaled such that  $\Delta = \tau\Delta$ , where  $\tau \geq 1$ . If the trial points do not lead to an improved solution such that  $\mathcal{L}_T({}^k\boldsymbol{\psi}_{i+k}^e) \geq \mathcal{L}_T(\boldsymbol{\psi}_i)$ , then the mesh is contracted such that  $\Delta = \delta\Delta$ , where  $0 < \delta < 1$ , and the solution candidate remains the same  $\boldsymbol{\psi}_{i+k} = \boldsymbol{\psi}_i$ . The algorithm terminates when the adaptation cost function reaches the given tolerance  $\mathcal{L}_T(\boldsymbol{\psi}_i) < \epsilon$ ; when  $\Delta < \Delta_{tol}$ , where  $\Delta_{tol}$  is the minimum mesh size; and when the algorithm reaches the maximum number of function evaluations  $i$ , as defined by user. An example of convergence of pattern search is illustrated in Fig. 5. Note that the lower right index indicates the total number of experiments (function evaluations) and the upper left index represents the trial point which is being evaluated within the mesh. Here, the algorithm would terminate in Fig. 5b, if  $\mathcal{L}_T(\boldsymbol{\psi}_7) < \epsilon$  is satisfied, or in Fig. 5d if  $\Delta < \Delta_{tol}$  is satisfied.

The mesh size and the mesh behavior influence the convergence of the PS algorithm and therefore need to be tuned. For the proposed framework, the number of points in the mesh should be minimum allowed, in order to minimize the number of experiments, since each evaluation of the objective function in PS corresponds to one experiment. Furthermore, if the objective function is more sensitive to one of the design parameters, the mesh should be scaled accordingly such that the step size is smaller in that direction. In this work, we keep the mesh size constant and scale the operation range covered by the excitation signal to a normalized space. That way, the PS operates in a uniform space with uniform mesh size. It is then translated to the real sampling space, where one of the parameters can be more sensitive. In addition, the algorithm can be set to evaluate each point on the mesh, or a greedy approach can be chosen where the mesh center is always moved to the first trial point with better solution. To avoid collecting the same data multiple times, the evaluated points are memorized. Further required modifications in the PS algorithm, such as initial mesh size and mesh behavior, are discussed in Sect. 3.



**Fig. 5** **a** Initial guess  $\psi_1$  is the candidate solution and the center of the mesh; three trial points are evaluated on the mesh, and the center is moved to the one with the smallest value of  $\mathcal{L}_T$  (shown as filled point); **b** the mesh is scaled, in this case by 1 and new trial points are evaluated; **c** better solution is not found by evaluating trial points; the candidate solution is kept; **d** the mesh is contracted; in this case it is halved, and new points are evaluated

### 3 Results and discussion

In Sect. 3.1, we demonstrate the proposed framework on an example of Duffing oscillator. This nonlinear dynamic system is commonly used for benchmarking data-driven modeling methods, such as in [32]. The proposed framework is then compared to several unsupervised sampling methods in Sects. 3.1.3 and 3.1.4. In Sect. 3.2, as an engineering application example, we show how the modeled force-state map of a nonlinear TMD can be integrated and used in the simulation of a two-story shear frame. All examples presented in this section are simulated with MATLAB [33].

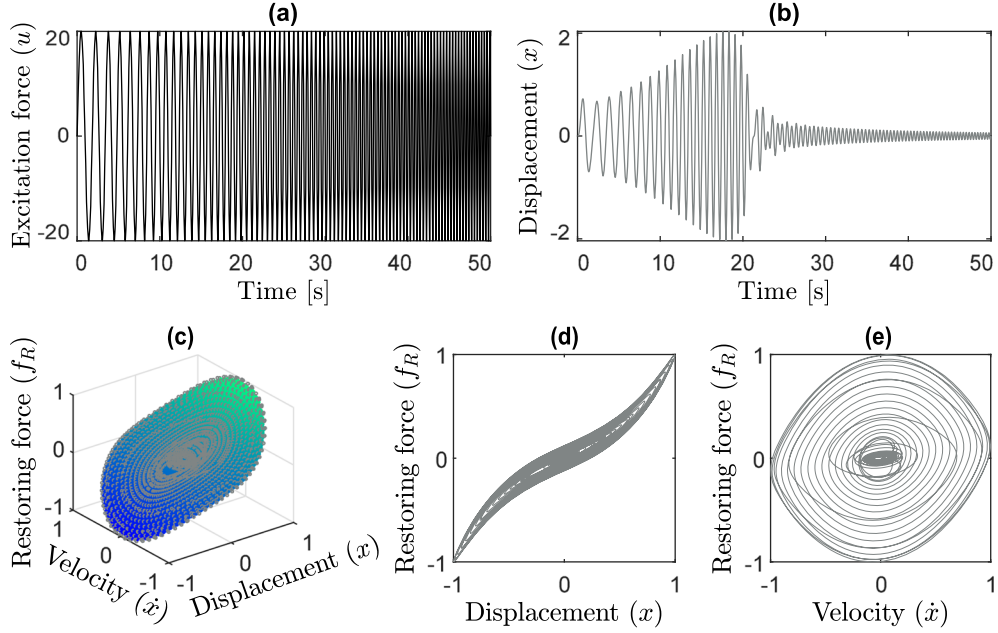
#### 3.1 Duffing oscillator

The restoring force of Duffing oscillator can be expressed in continuous form as:

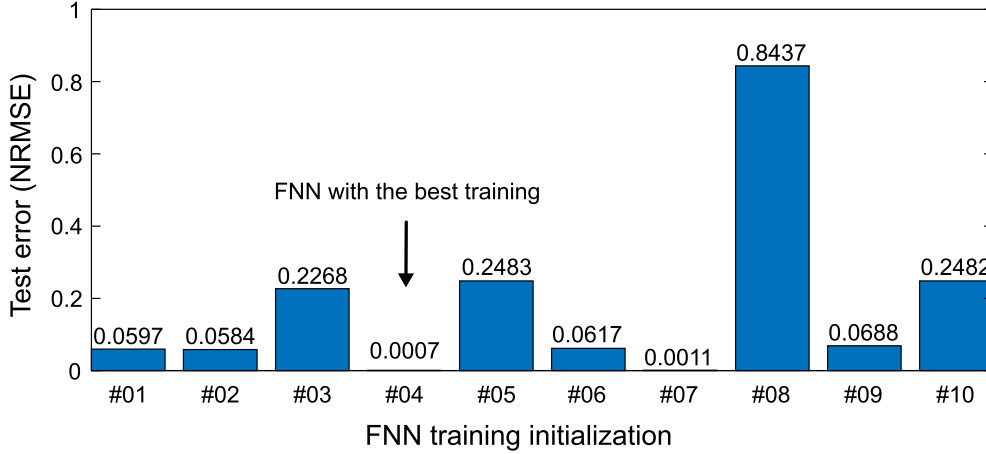
$$f_R(x, \dot{x}) = \alpha \dot{x} + \beta x + \gamma x^3, \quad (5)$$

where we assume a mass normalized system with  $m = 1$ ; damping coefficient  $\alpha = 1.25$ ; linear stiffness coefficient  $\beta = (2\pi)^2$ ; and nonlinear stiffness coefficient  $\gamma = 10$ . The calculated linear frequency of the system is 1 Hz. The system is simulated using 4<sup>th</sup> order Runge–Kutta method and with time step of 0.01 s.

The normalized force-state map which is modeled by FNN  $\mathcal{M}(\mathbf{\hat{x}}, \mathbf{x}, \boldsymbol{\theta})$  is depicted in Fig. 6. The response is computed by Eq. (5) for a *sine sweep* excitation and will be used to test the model throughout this section.



**Fig. 6** **a** Sine-sweep excitation signal; **b** analytically calculated displacement response of the Duffing oscillator; **c** normalized force-state map of the oscillator. The shading represents the values of the restoring force, ranging from  $-1$  (dark) to  $1$  (light); **d** normalized restoring force versus displacement plot; **e** normalized restoring force versus velocity plot

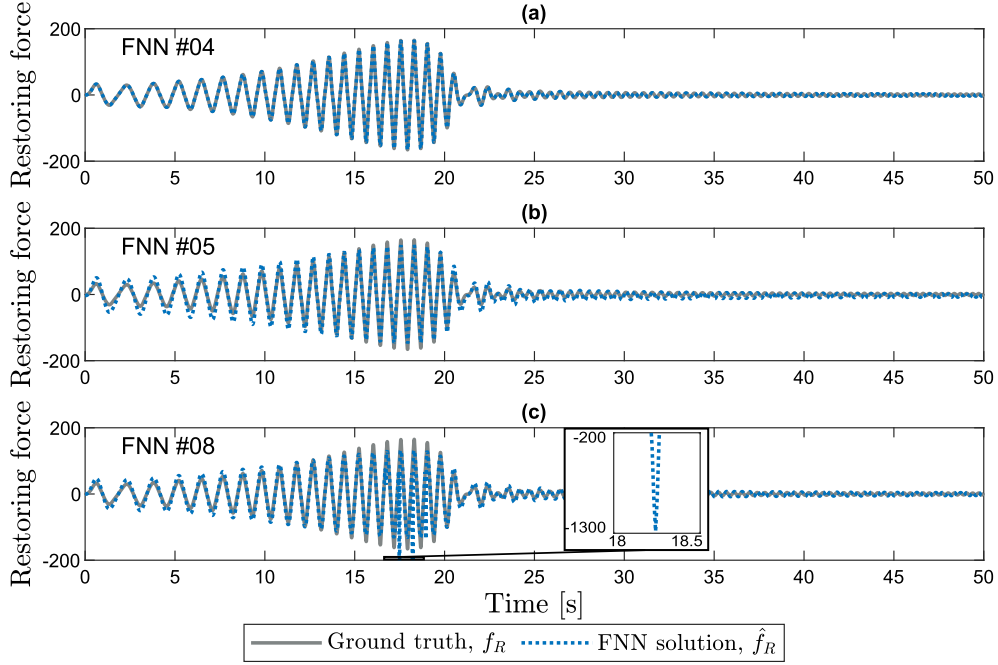


**Fig. 7** Same data set is used for the training of feedforward neural network (FNN) models with different initialization, yielding different test errors; using the model ensemble allows to choose the model with the best training

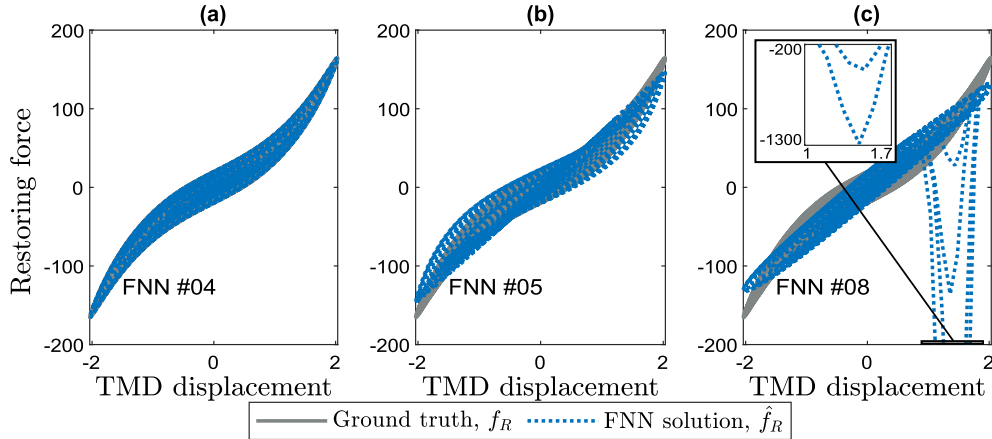
It is evident that the mapping is nonlinear, where in case of a linear system the force-state map would be a flat plane. However, it is important to reiterate that data cannot be directly sampled from this surface as the system state is not readily available. Therefore, data samples need to be collected by applying a dynamic excitation on the system, without a known outcome of the states and the corresponding restoring force. The sampling is therefore done in a separate space, and it is further illustrated in Sect. 3.1.3.

### 3.1.1 Choice of feedforward neural network configuration

An FNN with two hidden layers, each with two neurons, is chosen to model the force-state map. It is possible to use a larger architecture for modeling this system, such as in [34]; however, this would introduce larger model variance and therefore make the problem noisier. Since a model ensemble is used to eliminate this noise, there would be no significant effects in the efficacy of the proposed framework except increased computational time required for training. We use *hyperbolic tangent* activation functions and *mean squared error* as the loss



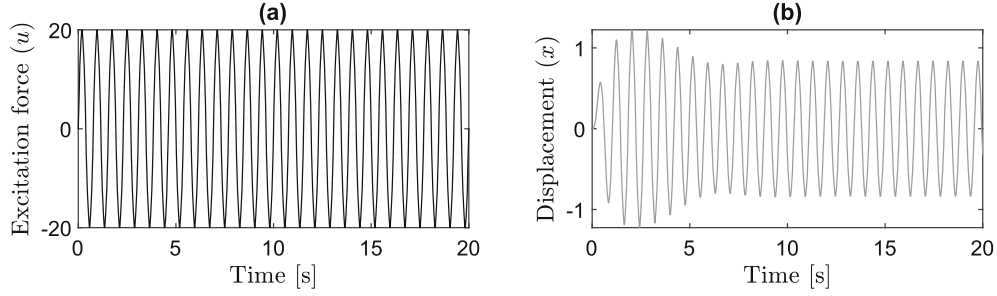
**Fig. 8** Restoring force time histories showing the performance of the FNN model on the test set with different training success, corresponding to the NRMSE of 0.0007 (a), 0.2483 (b) and 0.8437 (c), (cf. Fig. 7)



**Fig. 9** Force versus displacement plots showing the performance of the FNN model on the test set with different training success, corresponding to the NRMSE of 0.0007 (a), 0.2483 (b) and 0.8437 (c), (cf. Fig. 7)

function. The model is trained using the *Levenberg–Marquardt* training algorithm with *trainlm* function in MATLAB and its default settings. Before training, 3% noise is added to the restoring force to enhance the training and simulate possible noise contamination coming from measurements.

A model ensemble is used as described in Sect. 2.4 with 10 model members. Each member model of the ensemble has the same structure and hyperparameters, but different training initialization, such as initial model parameters. During the training, 80% of the data is used as the training data set and 20% of the data is used as the validation data set. The model performance is evaluated with normalized root mean square error (NRMSE). The necessity to use the model ensemble is justified by training the model several times with the same training data set and different training initialization. From the results shown in Fig. 7, it can be seen that the variance in the results is vast, where the model with the best performance has the NRMSE of 0.0007 and the one with the worst performance has the NRMSE of 0.8437 which is orders of magnitude higher.



**Fig. 10** **a** An example of the sine excitation signal used to obtain the training data. Excitation amplitude and frequencies are  $u_0 = 20$  and  $\Omega = 1.3$ , respectively; **b** Analytically calculated corresponding displacement response of the Duffing oscillator

Furthermore, in order to visualize the impact of different training performances, Figs. 8 and 9 show the performances of the FNN models #04, #05 and #08 on the sine-sweep test excitation signal (cf. Fig. 6a) as time history and force versus displacement plots, respectively.

### 3.1.2 Choice of training and test data sets

20-s-long *sine signals* are used as the excitation signals to generate the training data sets. In Fig. 10a, an example sine signal is depicted. The signal parameters are  $\psi = [u_0 \ \Omega]^\top$ , where  $u_0$  is the amplitude and  $\Omega$  is excitation frequency. It is noteworthy to mention that the response data are assumed to cover transient characteristics of the oscillator as shown in Fig. 10b. Consequently, the importance of the amplitude identification is not as high as in the case of a steady-state response. However, it should be emphasized that in the considered problem setting the user is assumed to have no prior knowledge about the response behavior of the investigated specimen.

Different and potentially more powerful excitation signals, such as pseudo-random binary signals (PRBS) and amplitude modulated pseudo-random binary signals (APRBS) [7,15], could also be used to obtain the training set. However, we prefer to introduce the methodology with sine signals, both due to pedagogical reasons and to obtain a fair comparison between the proposed method and unsupervised methods by avoiding any randomness coming from the signal itself, such as in case of a PRBS and APRBS excitation.

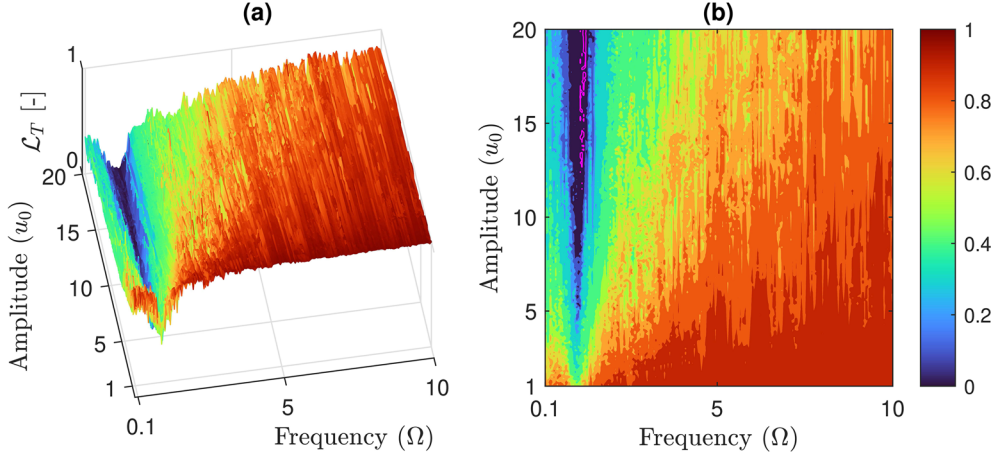
The test set is obtained with 50 s long *swept sine* excitation at the amplitude of 20 and the frequency range between 0.5 Hz and 3 Hz. The test set is also depicted in Fig. 6a. It is evident that the chosen test set covers the system's nonlinearity in its operation range, and therefore, it is representative of the system. It should be noted that multiple test sets could be used in this setup to further increase the model accuracy. We would like to also emphasize that the test set is kept constant and is not changed or updated during the proposed framework.

### 3.1.3 Sampling space

The sampling of data is done through the excitation signal parameter space. Hence, with the sine signal used for excitation, the sampling space is two dimensional. When a value indicating the quality of the sample  $\psi_i$  is assigned to each point in the sampling space, the adaptation cost function  $\mathcal{L}_T$  is obtained. As introduced before, the adaptation cost function represents the NRMSE of the model evaluated on the test set, after being trained with data extracted from the system's response excited by the excitation sine signal with the parameters  $\psi_i$ .

The error surface plotted in Fig. 11 shows the adaptation cost function  $\mathcal{L}_T$  as a function of excitation signal parameters  $\psi_i$  plotted in the operating range of amplitudes  $u_0 = [1 \dots 20]$  and excitation frequencies  $\Omega = [0.1 \dots 10]$  Hz. The error surface is produced by evaluating each set of parameters  $\psi_i$  on a uniform grid with the increment of 0.1 for amplitude and 0.05 for frequency.

From Fig. 11, it can be seen that the surface is convex with a global minimum, which confirms that an optimization problem can be formulated to find the best training set. At low excitation amplitude, the system response is linear and the best training data can be acquired with the excitation frequency corresponding to the linear natural frequency of the system around  $\Omega = 1$  Hz. Better models with lower error values can be generated after training with data of higher amplitudes, where the nonlinear characteristics of the system become also visible in the response. Here, the required excitation frequency for the optimal training is in a larger frequency band than in the linear case with low amplitude. It is noted that the evaluation of the error

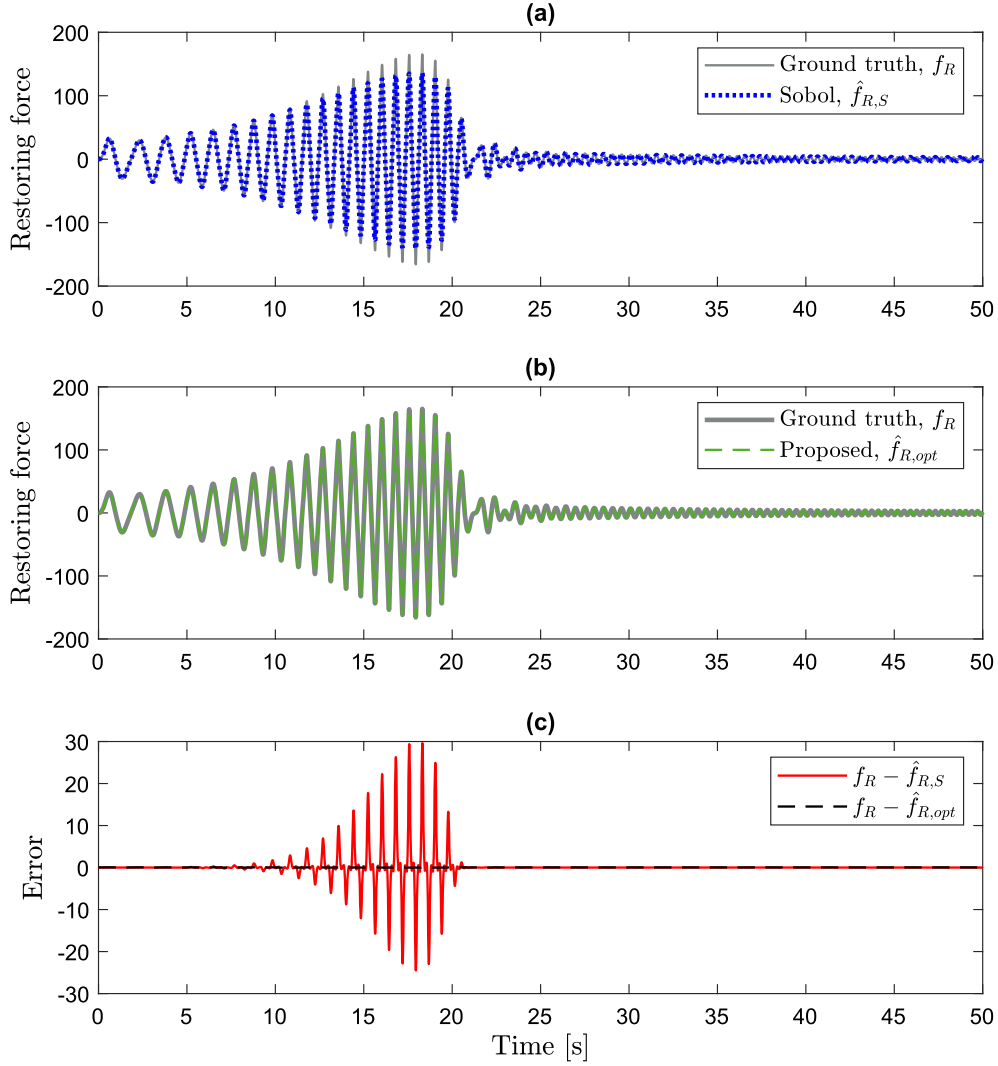


**Fig. 11** **a** Error surface  $\mathcal{L}_T$  produced by evaluating training sets on a uniform grid with respect to excitation signal parameters  $u_0$  and  $\Omega$ . The colorbar shows the normalized root mean square error; **b** contour plot corresponding to the error surface; the surface is convex and there are excitation parameter combinations which can lead to optimally trained high accuracy models of the Duffing oscillator

surface is not part of the proposed framework and is done solely to aid visualization. In the present problem, the error surface and the oscillator parameters are unknown and to be explored experiment by experiment. Our aim is to find the best excitation signal, which produces the best training data, after a few experiments.

The error surface is stochastic which justifies the proposed use of gradient-free optimization method. It is worth mentioning that it is expected that the error surface would change if different model architecture, hyperparameters, or training algorithm is used. In addition, the error surface is expected to be more stochastic as the mismatch between the model architecture and the system increases, since the training variance would increase. However, the shape of the error surface corresponds to system characteristics and is expected to remain the same for the Duffing oscillator. In addition, from authors' experience the error surface is expected to remain convex for other systems with different type of nonlinearity, such as piecewise linear (PWL) systems, since the system is expected to produce different responses under different excitation signals. Hence, there will always be excitation signal parameters, which produce more informative response of the system. However, the error surface would have a shape depending on the properties of the system, such as the natural frequency, and which excitation drives the system to cover the most informative states. For example, in case of a hardening (PWL) system, in the same setup as presented in this paper, a combination of training signal parameters that drives the system to cover both the initial and hardening stiffness would need to have a good coverage of the transition between them in order lead to the optimum training data set. Therefore, each combination of excitation signal parameters that fulfills the above-mentioned criteria would lead to an optimum training data set. For further systems with more complex material behavior, the proposed data generation framework may need to be adopted also to other neural network architectures. For instance, for memory systems, the long-short term memory (LSTM) networks can be utilized.

To reinforce the understanding of Fig. 11, we evaluate two different samples from the error surface. The first sample  $\psi_2 = [13.36 \ 0.62]^\top$  is used to produce the first training set  $\mathcal{D}_1 = \{\dot{\mathbf{x}}_1, \mathbf{x}_1, \mathbf{f}_{R,1}\}$ . The second sample  $\psi_1 = [19.26 \ 1.25]^\top$  is used to produce the second training set  $\mathcal{D}_2 = \{\dot{\mathbf{x}}_2, \mathbf{x}_2, \mathbf{f}_{R,2}\}$ , which corresponds to an optimum training set as defined in Sect. 2.2 and meets the desired tolerance  $\mathcal{L}_T = \epsilon = 0.01$ . Therefore, this set is denoted as  $\mathcal{D}_{\text{opt}}$ . The first training set  $\mathcal{D}_1$ , which does not meet the given tolerance, is randomly picked from the surface and has an error  $\mathcal{L}_T = 0.08$ . This randomly sampled training set is denoted as  $\mathcal{D}_S$ . When tested on a test set  $\mathcal{D}_T = \{\dot{\mathbf{x}}_T, \mathbf{x}_T, \mathbf{f}_{R,T}\}$ , they produce outputs  $\hat{\mathbf{f}}_{R,S}$  and  $\hat{\mathbf{f}}_{R,opt}$ , respectively, which are compared to the true value  $\mathbf{f}_{R,T}$  by calculating NRMSE to obtain  $\mathcal{L}_T$ . The effect of the tolerance error is visualized in Fig. 12, where the time histories corresponding to the restoring forces  $\hat{\mathbf{f}}_{R,S}$  and  $\hat{\mathbf{f}}_{R,opt}$  are depicted. It is therefore evident that two training sets, having the same amount of data, contain different amount of information used for model training. Furthermore, from Fig. 12c it can be seen that the chosen error tolerance  $\epsilon$  leads to  $\mathcal{D}_{\text{opt}}$  which allows to replicate linear and nonlinear behavior of the system accurately (cf. Fig. 13).

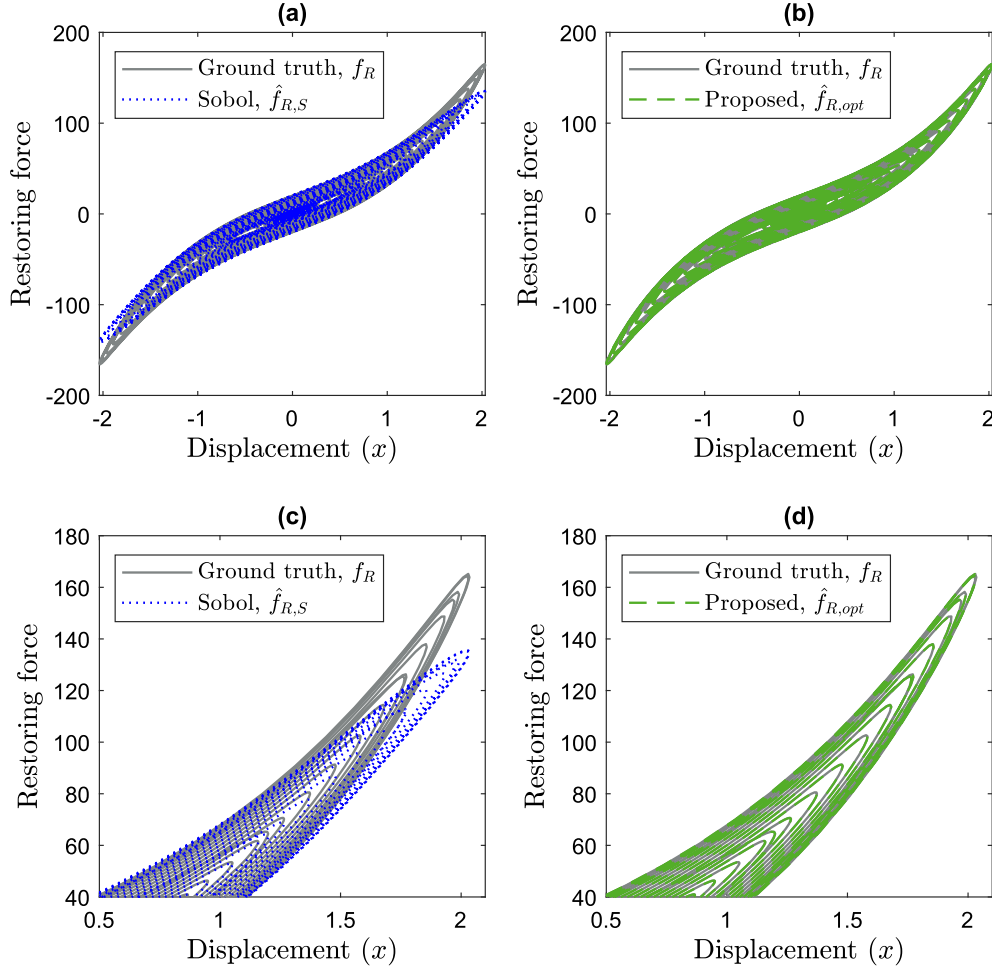


**Fig. 12** **a** Time history of the restoring force  $\hat{f}_{R,S}$  produced by the model trained with randomly sampled data set  $\mathcal{D}_S$  compared to the ground truth; **b** time history of the restoring force  $\hat{f}_{R,opt}$  produced by the model trained with an optimum data set  $\mathcal{D}_{opt}$  compared to the ground truth; **c** errors showing that  $\mathcal{D}_S$  and  $\mathcal{D}_{opt}$  lead to different quality of the model, although they have the same size

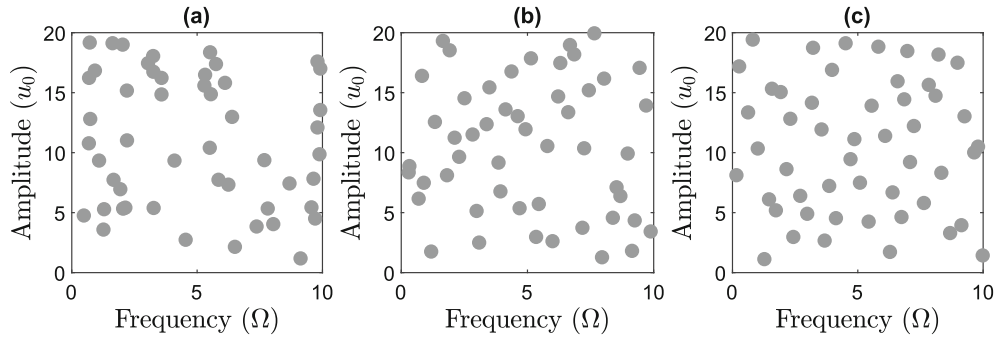
### 3.1.4 Investigated unsupervised sampling methods

We take three different unsupervised sampling methods as a reference to compare our method with: *uniform random sampling*, *Latin hypercube sampling* (LHS) and *Sobol sequence sampling*. In the parameter space, with each method a maximum of 50 points are sampled as depicted in Fig. 14. In each experiment campaign, 10, 20, 30 and 50 points are sampled, and each campaign is repeated for 1000 times in order to get statistically viable results, since in each instance of these methods the distribution of points is different. This corresponds to a total of 4000 experimental campaigns for each sampling method.

For each number of points, it is recorded how many times the optimum data set  $\mathcal{D}_{opt}$  is found. For example, for the *uniform random* method with 10 points, the simulation is run 1000 times and it is recorded every time the optimum data set  $\mathcal{D}_{opt}$  is found. Then, it is repeated with 20, 30, and 50 sampling points expecting that the results would improve as the number of samples is increased. The same is done for other sampling methods.



**Fig. 13** **a** Restoring force versus displacement plot produced by the model trained with a data set  $\mathcal{D}_S$  obtained with Sobol sequence, compared to the ground truth; **b** restoring force versus displacement plot produced by the model trained with an optimum data set  $\mathcal{D}_{opt}$  compared to the ground truth; **c** enhanced view of the plot shown in **a**; **d** enhanced view of the plot shown in **b**



**Fig. 14** Parameters sampled by **a** Uniform random sampling; **b** Latin hypercube sampling; and **c** Sobol sequence sampling

**Table 1** Pattern search simulation parameters used in the study

| Parameters         | Values                         |
|--------------------|--------------------------------|
| Initial mesh size  | 8                              |
| Poll method        | Minimal basis set (cf. Fig. 5) |
| Mesh tolerance     | 0.01                           |
| Scale mesh         | No                             |
| Use complete poll  | No                             |
| Function tolerance | 0.01                           |
| Use cache          | Yes                            |

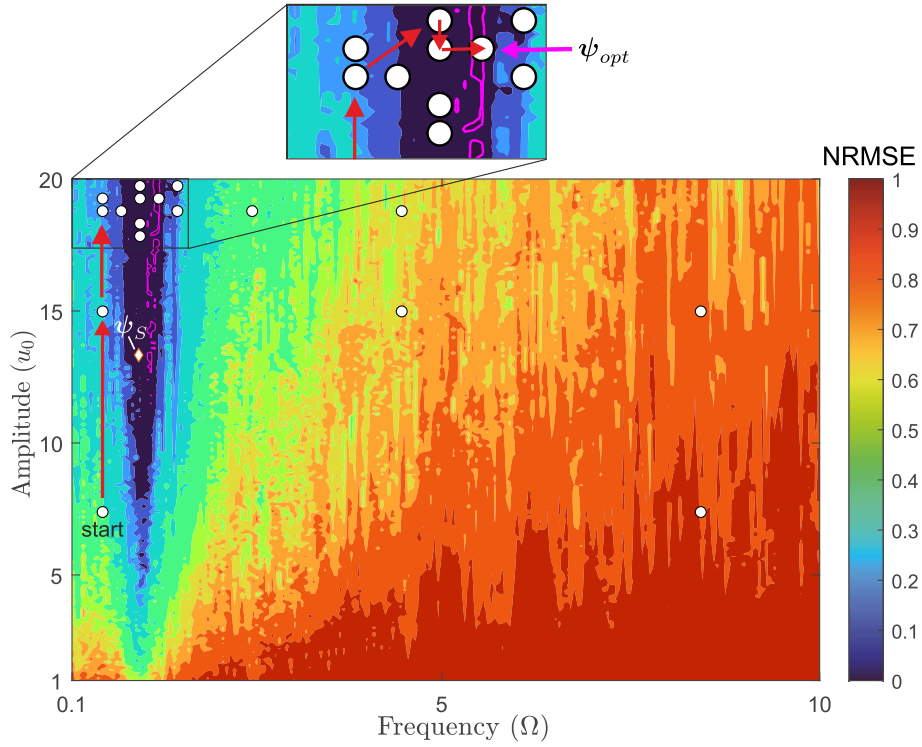
### 3.1.5 Adaptive data generation

The implementation of the PS optimization algorithm is done using function *patternsearch* in MATLAB. The algorithm objective function has the excitation signal amplitude and frequency as inputs and NRMSE as the output. When the objective function receives the excitation signal parameters, it creates the sine signal with these parameters and performs the numerical simulation of the response and the training of the ensemble. The ensemble is then evaluated, and the NRMSE of the model with the best performance is taken as the output. The first sample is chosen randomly, and every consequent sample is chosen by the PS algorithm. Similar to unsupervised methods, the proposed framework is conducted 1000 times with different starting points for the PS algorithm. The detailed simulation parameters are shown in Table 1. It is noted that normalizing the sampling space for PS and choosing a large starting mesh result in performance improvement. The modified PS finds the optimum training data set on average after 19 experiments, while the default algorithm finds it on average with 23 experiments.

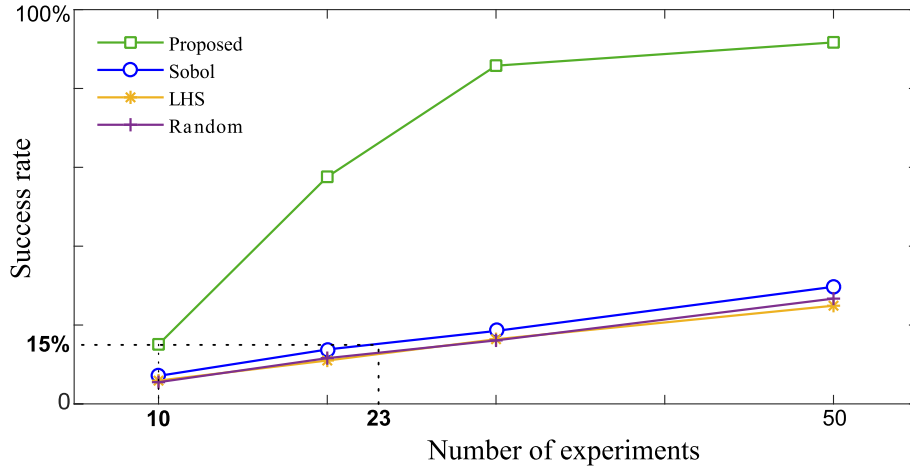
An example of the converged solution is shown in Fig. 15 in which it takes PS algorithm 17 experiments to find the optimum data set. For the sake of fair comparison, a solution found by Sobol sequence with the same number of experiments was plotted and indicated as  $\psi_S$ . These two examples will be used in the following sections for further illustrative examples. From the convergence of the proposed framework shown in Fig. 15, it can be seen that the algorithm starts with a random point which is far from the global minimum, and as it approaches the global minimum, it samples more densely in region of global minimum. This result shows that the collection of the dynamic data for training of the model is converted to the optimization problem in static sampling space. It is important to mention that this is only possible due to the assumption that there exists an optimal training data set  $\mathcal{D}_{\text{opt}}$  which can be generated using the sine excitation with the parameters  $\psi$ . If this assumption is not to hold true, the optimal training data set  $\mathcal{D}_{\text{opt}}$  would need to be built at each iteration. In that case, the error surface shown in Fig. 11 would evolve with each added batch of data to the training set  $\mathcal{D}$ . This is theoretically still possible with the proposed framework, where the PS algorithm would be used in the context of the dynamic function optimization. However, these goals are outside of the scope of this paper as they add significant degree of complexity to the proposed framework.

### 3.1.6 Comparison with unsupervised sampling methods

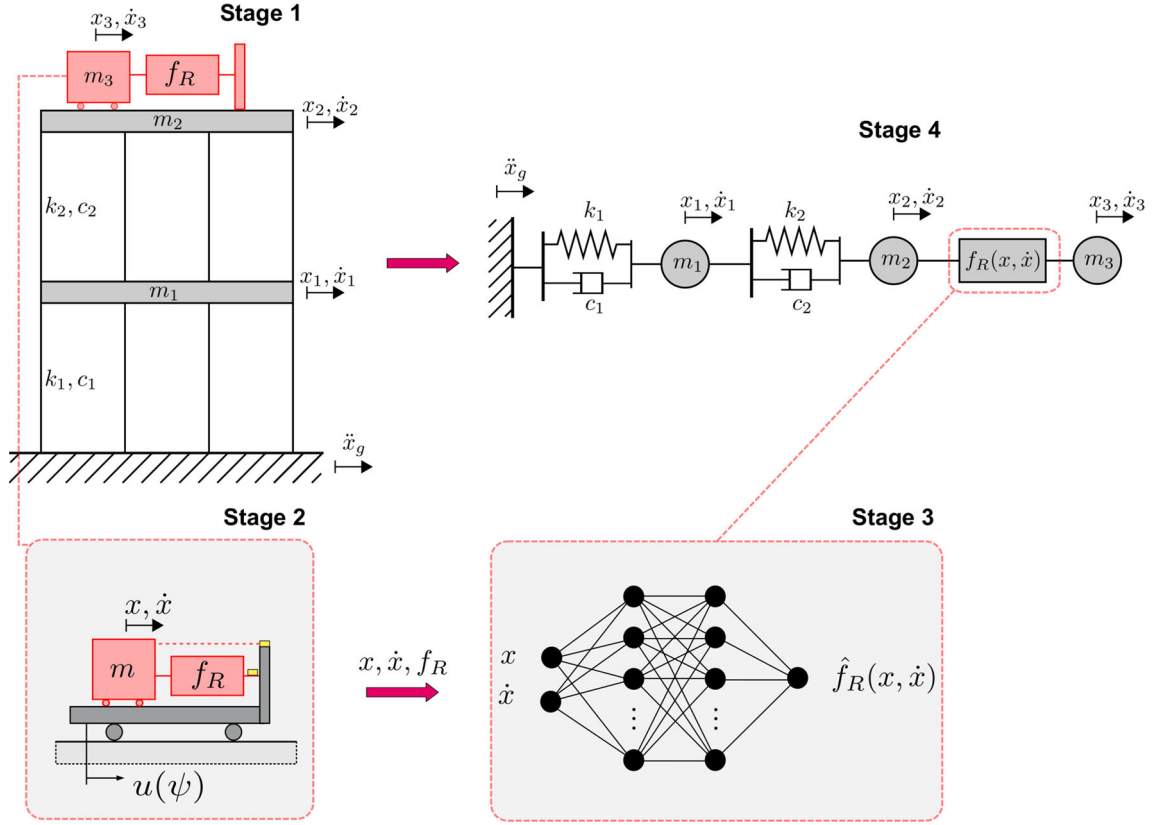
To compare the proposed adaptive sampling and the unsupervised sampling methods (uniform random sampling, Latin hypercube sampling (LHS), Sobol sequence sampling), we record the number of times the optimum training data set is found in 1000 simulations with a prescribed number of experiments (10–50). The results are presented in Fig. 16, where it can be seen that after 10 experiments, the proposed method finds the optimum training data set 151 times, which is equivalent to 15% success rate. The best of the unsupervised methods Sobol sequence requires at least 23 experiments to reach the same training success. The difference in performance increases as the number of samples is increased, and becomes pronounced for 50 experiments, where the proposed framework finds the optimum data set with almost 100% success rate, while the best of unsupervised methods Sobol sequence can reach at most 40% success. It is noted that random sampling and LHS have similar performance while Sobol sampling has a slightly better performance among unsupervised methods. It is expected that the difference between LHS and random sampling would increase in favor of LHS with an increase in dimensions of the sampling space. This is due to LHS sampling having equal space filling properties in all dimensions.



**Fig. 15** One example of convergence of the proposed framework showing that the optimum excitation parameters  $\psi_{opt}$  (encircled by the magenta line) which produce the optimum training data set  $\mathcal{D}_{opt}$  are found after 17 experiments; arrows point from the current point with the best solution to the next sample with better solution until  $\psi_{opt}$  is found, while other points are algorithm's exploratory evaluations; point indicated by  $\psi_s$  is not evaluated by the proposed framework and indicates the best solution found by Sobol sequence containing 17 samples



**Fig. 16** Plot showing the number of experiments required to achieve a certain success rate for each method; the proposed framework clearly outperforms the unsupervised methods since its success rate is more than twice higher for 10 experiments and continues to increase with the number of experiments



**Fig. 17** Two-story shear frame with a nonlinear TMD modeled by the proposed framework

**Table 2** System parameters used in the simulation

|           | Parameter  | Value      | Explanation                            |
|-----------|------------|------------|----------------------------------------|
| Structure | $m_1, m_2$ | 15         | Mass of rigid floors                   |
|           | $c_1, c_2$ | 1.25       | Damping coefficient of floor columns   |
|           | $k_1, k_2$ | 250        | Stiffness coefficient of floor columns |
| TMD       | $m_3$      | 1          | Mass of the TMD                        |
|           | $\alpha$   | 1.25       | Damping coefficient                    |
|           | $\beta$    | $(2\pi)^2$ | Stiffness coefficient                  |
|           | $\gamma$   | 10         | Nonlinear stiffness coefficient        |

### 3.2 Two-story shear frame with a nonlinear tuned mass damper

To demonstrate how the modeled restoring force can be integrated in the analysis, we consider a two-story shear frame with a nonlinear TMD on the top floor, cf. Fig. 17. The system is subjected to ground acceleration  $\ddot{x}_g$  and the corresponding equations of motion are written as

$$-m_1\ddot{x}_g(t) = m_1\ddot{x}_1 + k_1x_1 + k_2(x_1 - x_2) + c_1\dot{x}_1 + c_2(\dot{x}_1 - \dot{x}_2), \quad (6a)$$

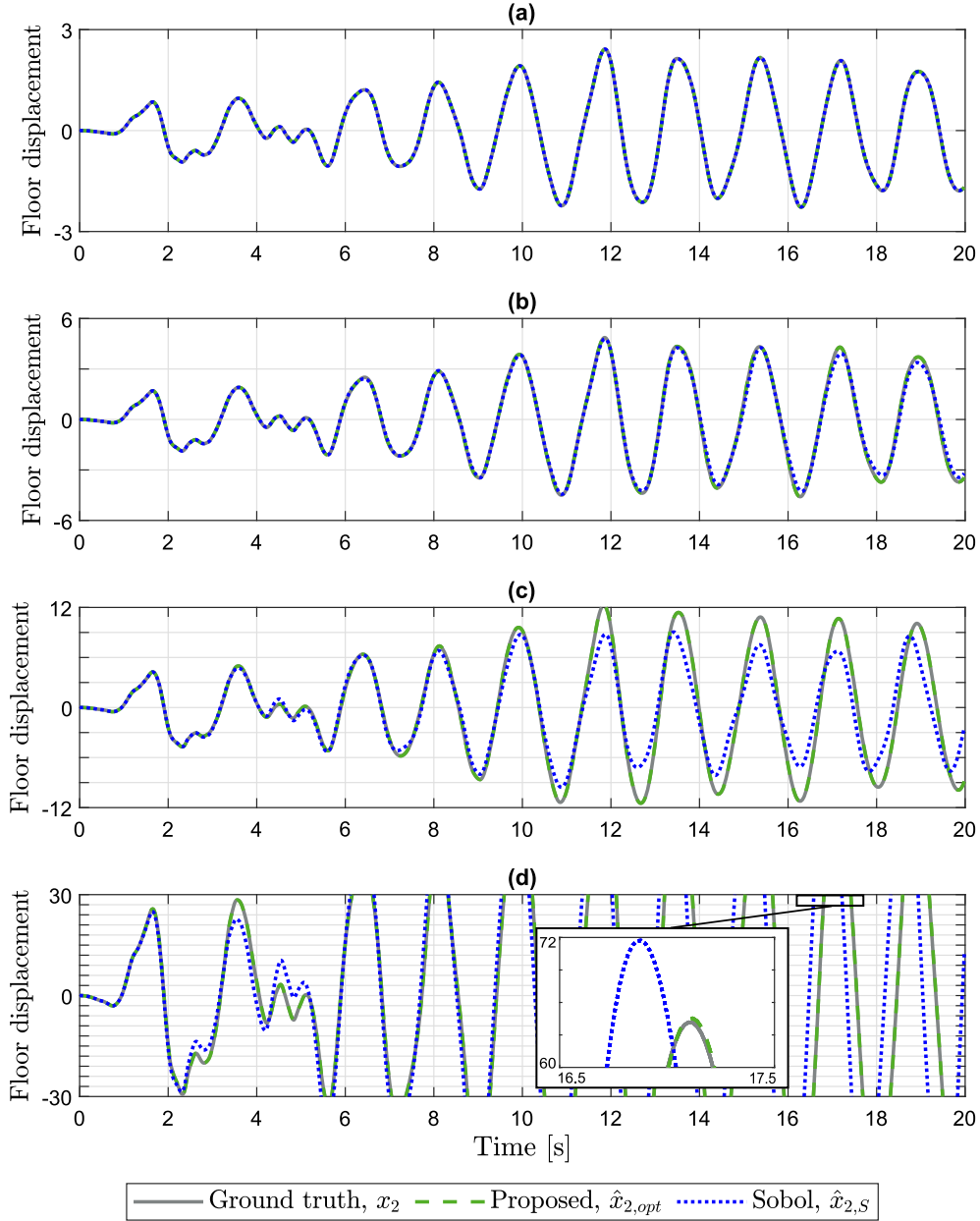
$$-m_2\ddot{x}_g(t) = m_2\ddot{x}_2 - k_2(x_1 - x_2) - c_2(\dot{x}_1 - \dot{x}_2) - f_R, \quad (6b)$$

$$-m_3\ddot{x}_g(t) = m_3\ddot{x}_3 + f_R, \quad (6c)$$

where  $f_R$  is the restoring force of the TMD including both linear and nonlinear parts and is expressed as

$$f_R = \beta(x_3 - x_2) + \gamma(x_3 - x_2)^3 + \alpha(\dot{x}_3 - \dot{x}_2). \quad (7)$$

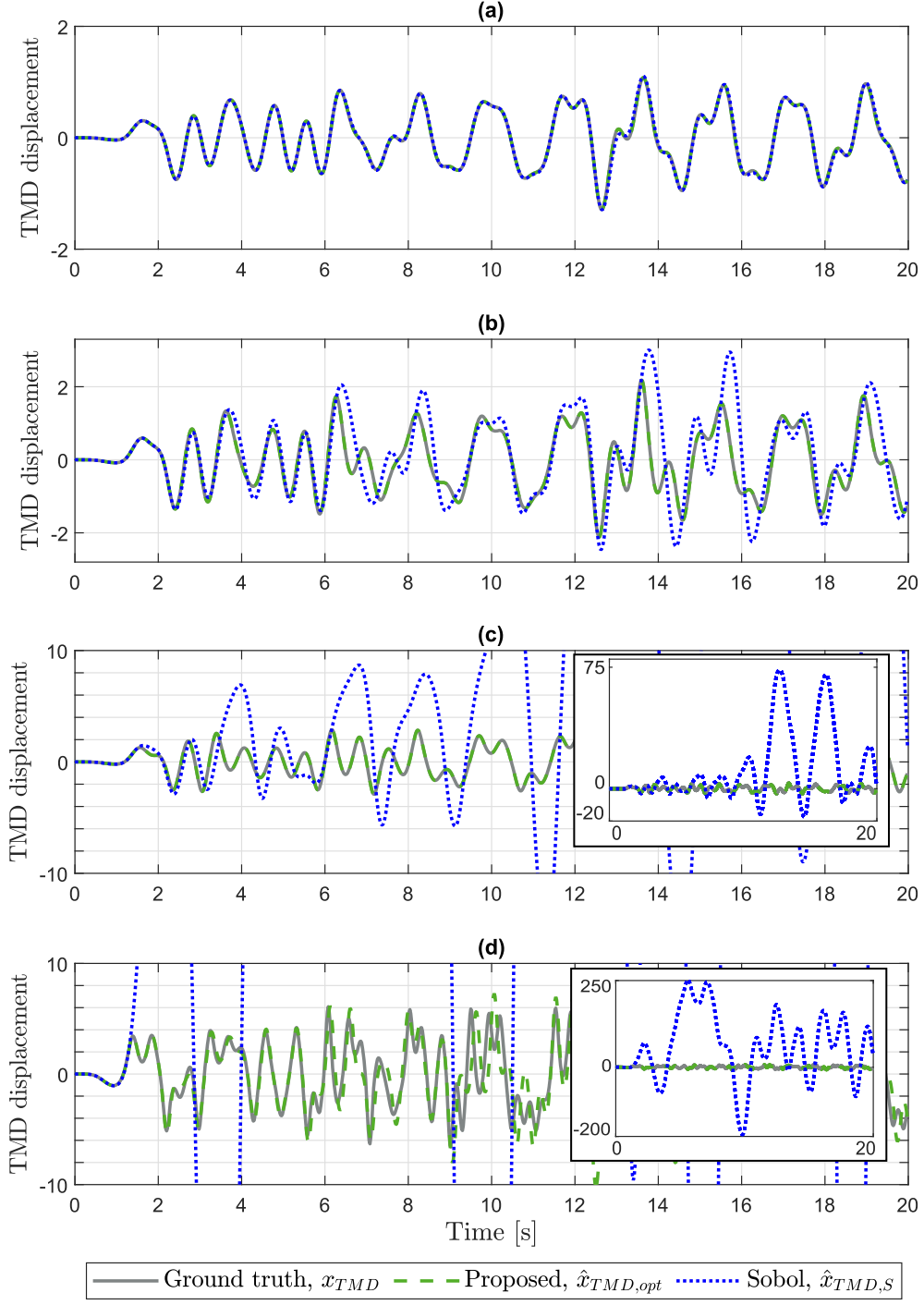
The parameters of the nonlinear TMD correspond to that of the Duffing oscillator with the restoring force described in Eq. (5). Both the parameter values of the structure and TMD are shown in Table 2. The structure's



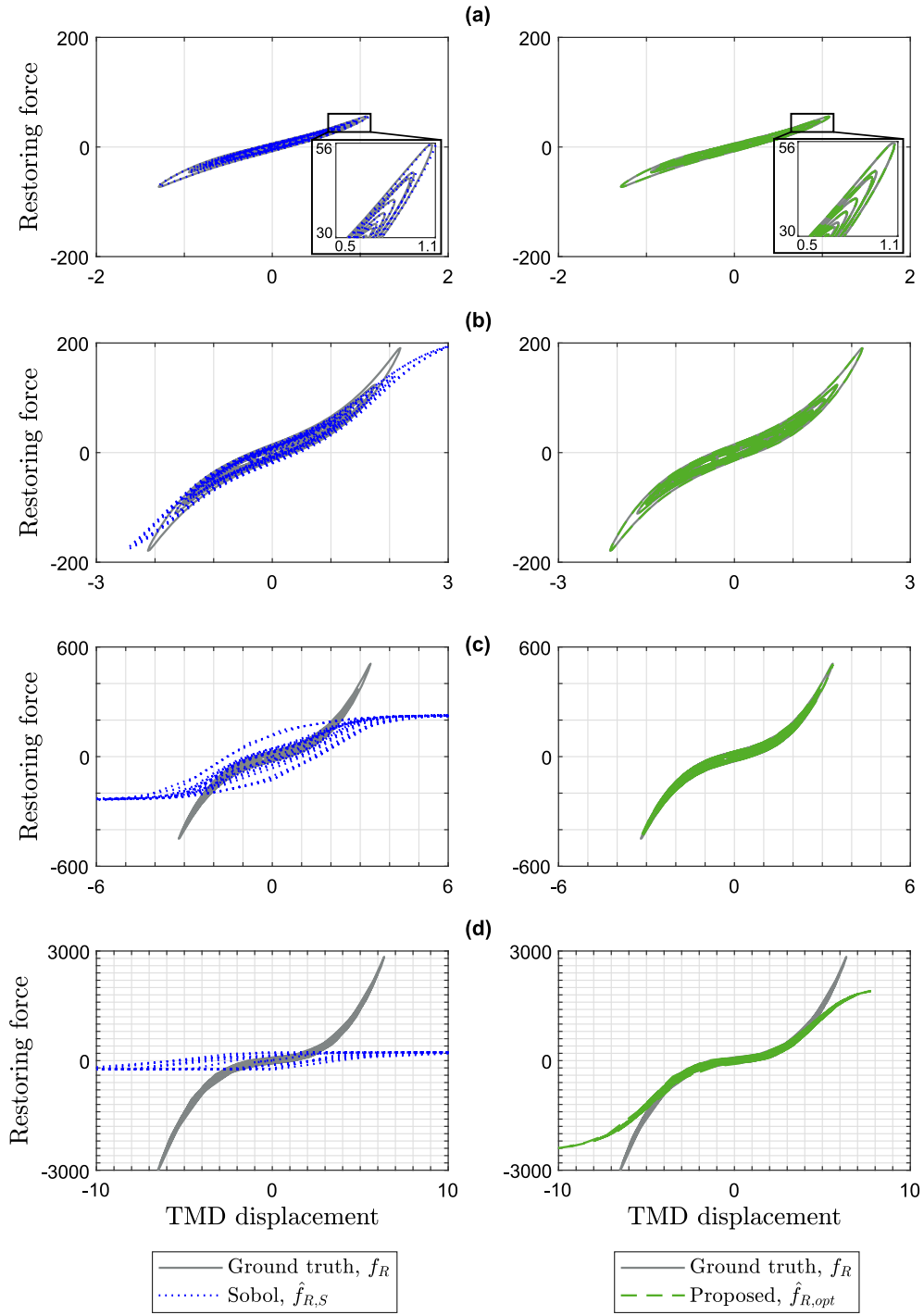
**Fig. 18** Plots showing the displacement responses of the 2nd floor using an FNN trained with the data sets obtained with proposed framework ( $x_{2,opt}$ ) and Sobol sequence ( $x_{2,S}$ ) when the excitation signal is scaled by  $\phi = 10$  (a), 20 (b), 50 (c) and 300 (d) times

first and second natural frequencies are 0.40 Hz and 1.05 Hz accordingly. The TMD is assumed to control the second natural frequency in this example. The structure is excited with the North–South component of the *El Centro Earthquake* recorded at the Imperial Valley Irrigation District substation in El Centro, California, on May 18, 1940. The earthquake is scaled by  $\phi = 10, 20, 50$  and 300 times to excite the system and to show the limits of the proposed framework and the competitive methods.

As shown in Fig. 17, the numerical example shows an engineering application carried out in four stages. In the *first stage*, the structure's response is computed as ground truth using Eqs. (6–7) and the parameters from Table 2, cf. Fig. 18. The *ode45* solver is used for this purpose. For the sake of brevity, only the displacement response of the second floor is depicted. In the *second stage*, the restoring force of the TMD is modeled based on the results obtained in Sect. 3.1.5 using the proposed data generation framework and the Sobol sequence corresponding to data sampling as depicted in Fig. 15. In the *third stage*, the obtained TMD model



**Fig. 19** Plots showing the displacement responses of the TMD using an FNN trained with the data sets obtained with proposed framework ( $\hat{x}_{TMD,opt}$ ) and Sobol sequence ( $\hat{x}_{TMD,S}$ ) when the excitation signal is scaled by  $\phi = 10$  (a), 20 (b), 50 (c) and 300 (d) times



**Fig. 20** Restoring force versus displacement plots of the TMD using an FNN trained with the data sets obtained with proposed framework ( $\hat{f}_{R,opt}$ ) and Sobol sequence ( $\hat{f}_{R,S}$ ) when the excitation signal is scaled by  $\phi = 10$  (a), 20 (b), 50 (c) and 300 (d) times

is implemented in the numerical simulation of the structure, where it is used instead of Eq. (7). Finally, in the *fourth stage* the structural response to the earthquake is simulated and compared to the ground truth obtained in first stage.

Figure 18 shows the displacement time histories of the second floor. Figure 19 shows the TMD displacements, where the derogation in the model performance is more visible. We observe that the FNN model obtained by the proposed data generation framework can properly model the restoring force of the TMD for the scaling factors  $\phi = 10, 20$  and  $50$ . Hence, there is no divergence between the modeled response and the ground truth. The first deterioration in the performance of the model starts with  $\phi = 300$ , which is far beyond its design range corresponding to  $\phi = 20$ . In contrast, the FNN model trained with the training set found by Sobol sequence does not perform well even for  $\phi = 20$ , and its performance continues to deteriorate sharply for  $\phi = 50$  and  $\phi = 300$ , as it can be seen from Figs. 18 and 19.

Figure 20 shows the restoring force over TMD displacement. The generalization capability of the FNN model obtained by the proposed framework exceeds the model obtained by Sobol sequence. If the restoring force of the Sobol sequence model is used in the simulation, the model is not able to represent the nonlinear characteristics of the TMD where the hardening effect is more pronounced. This leads to a vast overestimation of the displacements and an underestimation of the restoring forces. Hence, this result confirms the advantages of using the information about the model performance in order to sequentially sample new points using optimization.

## 4 Conclusion

In this paper, we proposed an adaptive data generation framework for the testing-integrated modeling of nonlinear dynamic systems using machine learning. The focus of the study was on systems where the mass cannot be decoupled from the restoring force, hence requiring sampling of the dynamic data. The proposed framework yields an FNN model of nonlinear restoring force by sequentially evaluating the model performance on a given test data and providing a new set of excitation signal parameters in order to get more informative data in the next iteration. Accordingly, the collection of the dynamic data is converted to an optimization problem in a static sampling space defined by the excitation signal parameters. Hence, the main advantages of the proposed framework are the following:

- The proposed framework converts the dynamic data sampling into a static optimization problem which enables the use of powerful conventional optimization algorithms.
- The proposed framework is not tied to a specific machine learning model choice, allowing the user to choose the most suitable model class.
- By using a test set which covers the operating range of the modeled system, the final modeling error also provides the model confidence level. This assumption comes from the focus on structural dynamics applications, where such experiments are already being conducted for performance assessment, such as with vibration control devices.

The proposed framework was validated numerically on an example of Duffing oscillator and compared to three commonly used unsupervised sampling methods. It outperformed the unsupervised methods, having a success rate of finding the optimum training set twice as high. Furthermore, an engineering application was illustrated where the response of a two-story shear frame with a nonlinear tuned mass damper was simulated using the FNN model of the damper's restoring force obtained both by the proposed framework and by the unsupervised methods. This example highlighted the capability of the proposed framework to extract the full information about the nonlinear response of the system.

The application of the proposed data generation framework to other testing configurations, such as damping testing systems, with direct restoring force identification, and the adoption of corresponding machine learning methods will be the focus of future research.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

**Funding** Open Access funding enabled and organized by Projekt DEAL.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Ethics approval** Not applicable.

**Consent to participate** Not applicable.

**Consent for publication** Not applicable.

**Data availability** Data are readily available from the authors at request.

**Author Contributions** PM took part in conceptualization, data curation, formal analysis, methodology, investigation, software, validation, visualization, writing—original draft; OA involved in conceptualization, methodology, writing—review and editing, supervision.

## References

- Di Matteo, A., Furtmüller, T., Adam, C., Pirrotta, A.: Optimal design of tuned liquid column dampers for seismic response control of base-isolated structures. *Acta Mech.* **229**(2), 437–454 (2018). <https://doi.org/10.1007/s00707-017-1980-7>
- Loh, C.-H., Loh, K.J., Yang, Y.-S., Hsiung, W.-Y., Huang, Y.-T.: Vibration-based system identification of wind turbine system: vibration-based system identification of turbine blades. *Struct. Control Health Monit.* **24**(3), 1876 (2017). <https://doi.org/10.1002/stc.1876>
- Casciati, F., Faravelli, L.: Experimental investigation on the aging of the base isolator elastomeric component. *Acta Mech.* **223**(8), 1633–1643 (2012). <https://doi.org/10.1007/s00707-012-0636-x>
- Sadati, S.M.S., Nobari, A.S., Naraghi, T.: Identification of a nonlinear joint in an elastic structure using optimum equivalent linear frequency response function. *Acta Mech.* **223**(7), 1507–1516 (2012). <https://doi.org/10.1007/s00707-012-0656-6>
- Ding, H., Kaup, A., Wang, J.-T., Lu, L.-Q., Altay, O.: Real-time hybrid simulation framework for the investigation of soil-structure interaction effects on the vibration control performance of shape memory alloys. *Eng. Struct.* **243**, 112621 (2021). <https://doi.org/10.1016/j.engstruct.2021.112621>
- Quaranta, G., Lacarbonara, W., Masri, S.F.: A review on computational intelligence for identification of nonlinear dynamical systems. *Nonlinear Dyn.* **99**(2), 1709–1761 (2020). <https://doi.org/10.1007/s11071-019-05430-7>
- Nelles, O.: Nonlinear System Identification: From Classical Approaches to Neural Networks, Fuzzy Models, and Gaussian Processes. Springer, Cham (2020). <https://doi.org/10.1007/978-3-030-47439-3>
- MacKay, D.J.: Information-based objective functions for active data selection. *Neural Comput.* **4**(4), 590–604 (1992). <https://doi.org/10.1162/neco.1992.4.4.590>
- Settles, B.: Active Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, vol. 18. Morgan & Claypool, San Rafael (2012). <https://doi.org/10.2200/S00429ED1V01Y201207AIM018>
- Nabian, M.A., Gladstone, R.J., Meidani, H.: Efficient training of physics-informed neural networks via importance sampling. *Comput. Aided Civ. Infrastruct. Eng.* **36**(8), 962–977 (2021). <https://doi.org/10.1111/mice.12685>
- Wu, C., Zhu, M., Tan, Q., Kartha, Y., Lu, L.: A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. *arXiv:2207.10289* (2022)
- Zhao, L., Li, Z., Caswell, B., Ouyang, J., Karniadakis, G.E.: Active learning of constitutive relation from mesoscopic dynamics for macroscopic modeling of non-Newtonian flows. *J. Comput. Phys.* **363**, 116–127 (2018). <https://doi.org/10.1016/j.jcp.2018.02.039>
- Belz, J., Bamberger, K., Nelles, O., Carolus, T.: Goal-oriented active learning with local model networks. *Int. J. Comput. Methods Exp. Meas.* **6**(4), 785–796 (2017). <https://doi.org/10.2495/CMEM-V6-N4-785-796>
- Khalid, M., Yusof, R., Joshani, M., Selamat, H., Joshani, M.: Nonlinear identification of a magneto-rheological damper based on dynamic neural networks. *Comput. Aided Civ. Infrastruct. Eng.* **29**, 3 (2014). <https://doi.org/10.1111/mice.12005>
- Gringard, M., Kroll, A.: On the parametrization of APRBS and multisine test signals for the identification of nonlinear dynamic TS-models. In: 2016 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–8 (2016). <https://doi.org/10.1109/SSCI.2016.7849984>
- Heinz, T.O., Nelles, O.: Iterative excitation signal design for nonlinear dynamic black-box models. *Procedia Comput. Sci.* **112**, 1054–1061 (2017). <https://doi.org/10.1016/j.procs.2017.08.112>

17. Hametner, C., Stadlbauer, M., Deregnaucourt, M., Jakubek, S., Winsel, T.: Optimal experiment design based on local model networks and multilayer perceptron networks. *Eng. Appl. Artif. Intell.* **26**(1), 251–261 (2013). <https://doi.org/10.1016/j.engappai.2012.05.016>
18. Buisson-Fenet, M., Solowjow, F., Trimpe, S.: Actively learning Gaussian process dynamics. In: *Learning for Dynamics and Control*, pp. 5–15 (2020). <https://doi.org/10.48550/arXiv.1911.09946>
19. Bull, L.A., Rogers, T.J., Wickramarachchi, C., Cross, E.J., Worden, K., Dervilis, N.: Probabilistic active learning: an online framework for structural health monitoring. *Mech. Syst. Signal Process.* **134**, 106294 (2019). <https://doi.org/10.1016/j.ymssp.2019.106294>
20. Bull, L., Worden, K., Manson, G., Dervilis, N.: Active learning for semi-supervised structural health monitoring. *J. Sound Vib.* **437**, 373–388 (2018). <https://doi.org/10.1016/j.jsv.2018.08.040>
21. Yuan, Y., Au, F.T., Yang, D., Zhang, J.: Active learning structural model updating of a multisensory system based on kriging method and Bayesian inference. *Comput. Aided Civ. Infrastruct. Eng.* (2022). <https://doi.org/10.1111/mice.12822>
22. Gardner, P., Dal Borgo, M., Ruffini, V., Hughes, A.J., Zhu, Y., Wagg, D.J.: Towards the development of an operational digital twin. *Vibration* **33**(3), 235–265 (2020). <https://doi.org/10.3390/vibration3030018>
23. Crawley, E.F., O'donnell, K.J.: Force-state mapping identification of nonlinear joints. *AIAA J.* **25**(7), 1003–1010 (1987). <https://doi.org/10.2514/3.9733>
24. Siddique, N.H., Adeli, H.: *Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks, and Evolutionary Computing*. Wiley, Chichester (2013)
25. Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. MIT Press, Cambridge (2016)
26. Worden, K.: *Nonlinearity in Structural Dynamics: Detection, Identification and Modelling*. CRC Press, Boca Raton (2019). <https://doi.org/10.1201/9780429138331>
27. Lai, Z., Mylonas, C., Nagarajaiah, S., Chatzi, E.: Structural identification with physics-informed neural ordinary differential equations. *J. Sound Vib.* **508**, 116196 (2021). <https://doi.org/10.1016/j.jsv.2021.116196>
28. Ding, H., Wang, J.-T., Altay, O., Lu, L.-Q.: Multilayer toroidal tuned liquid column dampers for seismic vibration control of structures. *Structures* **33**, 406–422 (2021). <https://doi.org/10.1016/j.istruc.2021.04.041>
29. Torczon, V.: On the convergence of pattern search algorithms. *SIAM J. Optim.* **7**(1), 1–25 (1997). <https://doi.org/10.1137/S1052623493250780>
30. Kolda, T.G., Lewis, R.M., Torczon, V.: Optimization by direct search: new perspectives on some classical and modern methods. *SIAM Rev.* **45**(3), 385–482 (2003). <https://doi.org/10.1137/S003614450242889>
31. Audet, C., Dennis, J.E.: Analysis of generalized pattern searches. *SIAM J. Optim.* **13**(3), 889–903 (2002). <https://doi.org/10.1137/S1052623400378742>
32. Li, S., Yang, Y.: A recurrent neural network framework with an adaptive training strategy for long-time predictive modeling of nonlinear dynamical systems. *J. Sound Vib.* **506**, 116167 (2021). <https://doi.org/10.1016/j.jsv.2021.116167>
33. MATLAB: Version 9.11.0 (R2021b). The MathWorks Inc., Natick, Massachusetts (2021)
34. Masri, S.F., Chassiakos, A.G., Caughey, T.K.: Identification of nonlinear dynamic systems using neural networks. *J. Appl. Mech.* **60**(1), 123–133 (1993). <https://doi.org/10.1115/1.2900734>