Engineering Applications of Artificial Intelligence 173 (2026) 114487

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Research paper

Active learning with neural ordinary differential equations for data-efficient modeling of tuned mass dampers

Pavle Milicevic [a], Sebastian Stemmler [b], Okyay Altay [a,c,*]

[a] RWTH Aachen University, Department of Civil Engineering, Chair of Structural Analysis and Dynamics, Mies-van-der-Rohe-Str. 1, 52074 Aachen, Germany
[b] RWTH Aachen University, Department of Mechanical Engineering, Institute of Automatic Control, Campus-Boulevard 30, 52074 Aachen, Germany
[c] University of Siegen, Department of Civil Engineering, Chair of Structural Analysis, Paul-Bonatz-Str. 9-11, 57076 Siegen, Germany

ARTICLE INFO

Keywords:
Active learning
Adaptive data generation
Design of experiments
Structural vibration control
Tuned mass dampers
Neural ordinary differential equations

ABSTRACT

The increasing use of structural vibration control devices in civil engineering necessitates efficient data-driven modeling to support rapid design and real-time applications, including digital twins. Performance testing of tuned mass dampers (TMDs) and their variants, such as tuned liquid column dampers (TLCDs), which are widely used in engineering practice, provides an excellent basis for developing and validating such models. However, comprehensive testing remains costly and time-consuming. This study proposes an active learning framework that autonomously optimizes experimental parameters to generate the most informative training data with minimal effort. The framework employs pattern search optimization to identify optimal test conditions based on model accuracy metrics. Neural Ordinary Differential Equations (Neural ODEs) are utilized to model response data, accurately capturing nonlinearities while eliminating the need for separate restoring-force measurements. The proposed approach extends the standard Neural ODE architecture to model forced vibrations and simultaneously estimate the often-unknown modal mass, addressing a fundamental limitation in TMD identification. Experimental validation is conducted on a TLCD using a shaking table, employing harmonic excitation for training and sine-sweep excitation for model error estimation, in accordance with standard performance testing protocols. The results demonstrate that the proposed framework achieves a 90% success rate with only 20 experiments, a 43% reduction compared to the 35 experiments required by conventional unsupervised sampling methods. Robustness is further validated through broadband noise testing, confirming generalization beyond training conditions. Finally, the developed model is applied to a numerical example, demonstrating its broader applicability to structural engineering problems.

1. Introduction

Growing environmental, economic, and architectural demands in civil engineering lead to slender, lightweight structures with low inherent damping, making them prone to vibrations. This creates an increasing demand for structural vibration control devices that provide supplementary damping and help maintain structural integrity and serviceability. This study focuses on tuned mass dampers (TMDs) and, in particular, on their variations, such as tuned liquid column dampers (TLCDs), which are widely used in civil engineering applications, including skyscrapers and wind turbines. Traditionally, analyzing such structures requires labor-intensive time-domain computations for multiple dynamic loading scenarios using large high-fidelity models. The complex damping mechanisms in these vibration control devices introduce nonlinearities, further increasing the complexity of these calculations. Therefore, efficient and accurate models are critically needed, representing a challenging task for traditional physics-based analytical models.

As an alternative, data-driven modeling can be employed, such as through various types of artificial neural networks (ANNs). Following supervised training, these methods can learn to replicate diverse response characteristics based on the provided data. Notably, vibration control devices undergo laboratory performance testing, both during manufacturing and as required by relevant codes, before being approved for structural use. This process presents an opportunity to develop data-driven models through a testing-integrated approach. However, current testing procedures typically do not produce a functional model, representing a gap that this paper aims to address.

The main challenge lies in reducing the amount of required training data to maintain the economic feasibility of physical testing. To address this challenge, this study integrates active learning with Neural

* Corresponding author at: University of Siegen, Department of Civil Engineering, Chair of Structural Analysis, Paul-Bonatz-Str. 9-11, 57076 Siegen, Germany.
E-mail address: okyay.altay@uni-siegen.de (O. Altay).

https://doi.org/10.1016/j.engappai.2026.114487
Received 11 March 2025; Received in revised form 4 February 2026; Accepted 11 March 2026
Available online 14 March 2026

Ordinary Differential Equations (Neural ODEs) to enable data-efficient modeling directly during testing phase. The main contributions of this paper are:

1. Extension of the Neural ODE architecture to accommodate cases involving unknown modal mass and forced vibrations.
2. Integration of the Neural ODE model with active learning to minimize the number of required experiments.
3. Experimental validation of the framework using a TLCD prototype.
4. Demonstration of the trained TLCD model within a practical structural application.

The structure of the paper is as follows: Section 2 summarizes the relevant literature. Section 3 introduces the proposed methodology, followed by Section 4, which presents the validation of the framework through experimental results for the TLCD. Section 5 demonstrates the application of the developed TLCD model in an engineering example. Finally, Section 6 summarizes the findings of the study.

2. Related work

TMDs and TLCDs are widely used to suppress undesired vibrations in civil structures; however, their nonlinear behavior complicates analytical modeling. A review of these devices, including their advanced versions and application examples, is provided in Altay (2021).

2.1. Data-driven modeling and design of experiments

Data-driven modeling has been increasingly used as an alternative to physics-based modeling in structural dynamics. Cunha et al. (2023) reviewed ANNs for response prediction in dynamic systems. These models rely on supervised training and require sufficiently informative datasets to generalize well. However, a key limitation of supervised learning approaches is their demand for large, high-quality labeled datasets, which can be prohibitively expensive to acquire in experimental settings. To reduce data acquisition cost, design of experiments (DOE) can be used to optimize sampling efficiency (Fisher, 1990). DOE encompasses a variety of methods, which are broadly classified into unsupervised (batch) sampling and adaptive (sequential) sampling methods. Unsupervised sampling includes methods such as random sampling and Latin Hypercube Sampling (LHS), where sampling points are chosen in advance. While computationally efficient, these methods do not leverage information gained during experimentation to guide subsequent data collection. In contrast, adaptive sampling methods sequentially optimize sampling strategies based on information gain from previous experiments. For example, Hametner et al. (2013) provided an approach for the identification of nonlinear dynamic systems using the Fisher information matrix as criteria for information gain. Although adaptive methods can achieve superior sampling efficiency, they typically require pre-existing models or surrogate functions to evaluate information content, which may not be available in early-stage testing scenarios.

2.2. Active learning for data-efficient model development

Active learning extends adaptive sampling in the field of machine learning where the model autonomously selects the most informative data for learning. By focusing on data points that are likely to yield the most significant improvements, this approach enables more efficient training with less labeled data, as described in Nelles (2020). Comprehensive reviews and applications of active learning are provided in Settles (2009, 2012), Ren et al. (2021). When integrated into testing environments, active learning can enable the automatic discovery of physical systems, as recently demonstrated in the context of wind tunnel experiments (Shields et al., 2023). Active learning strategies can be classified based on data availability scenarios: methods utilizing existing data pools, stream-based approaches for continuously arriving data, and query synthesis for generating new experimental data from scratch (Settles, 2012).

Recent work has further advanced active learning methodologies. Chen et al. (2025) proposed an uncertainty-guided Bayesian active learning framework for fault diagnosis, achieving up to 89% reduction in labeling effort for time-series data. While this demonstrates the power of uncertainty-based sampling, the computational overhead of Bayesian inference may limit scalability for large-scale dynamic systems. Sobot et al. (2024) developed a human-in-the-loop active learning approach for time-series electrical measurement data, incorporating expert confidence levels to mitigate labeling errors. Although effective for classification tasks, this framework's reliance on human experts for each labeling decision may not be practical for autonomous experimental testing environments. Samoa et al. (2024) introduced a unified active learning framework for graph-level regression tasks, demonstrating improved annotation efficiency for complex data structures. However, graph-based representations may not directly translate to time-series dynamic response modeling of mechanical systems.

In structural engineering and dynamics, Bull et al. applied active learning to aircraft monitoring (Bull et al., 2019), demonstrating effective pool-based sampling for structural health monitoring. This work was further improved by Hughes et al. through the elimination of sampling bias (Hughes et al., 2022). While these approaches successfully reduced labeling effort, they assume pre-existing data pools and do not address scenarios where data must be generated through physical experimentation. Kiani et al. presented another pool-based example for response modeling using ANNs, where the authors selected the most informative samples from an existing set of ground motion data (Kiani et al., 2020). The approach significantly reduced the number of required ground motion analyses while improving prediction accuracy compared to randomly selected data. However, the reliance on pre-existing ground motion databases limits applicability to newly manufactured devices requiring testing from scratch. Similarly, Zhang et al. sampled stiffness parameters from a predefined pool to develop a surrogate model (Zhang et al., 2023), and Gardner et al. employed stream-based active learning to update digital twins based on quantified model uncertainty (Gardner et al., 2020).

Using Kriging models, Abbiati et al. (2022) applied active learning to hybrid simulations, where the goal was to create a classifier to determine failure and safe states of structural components. The strength of Kriging-based approaches lies in their inherent uncertainty quantification, which naturally guides informative sampling. However, computational costs scale poorly with dataset size, limiting their applicability to large-scale testing programs. Again using Kriging, Yuan et al. (2023) introduced a response modeling framework that significantly improved identification rates while reducing required sample sizes by half. Despite these advantages, Kriging methods often struggle to capture highly nonlinear dynamics and require careful selection of kernel functions, which may not generalize well across different types of vibration control devices.

Active learning has also been integrated with Gaussian Process State-Space Models (GPSSMs) to discover system dynamics (Yu et al., 2021). The authors highlighted the distinction from static active learning problems, noting that in dynamic systems, only the inputs can be controlled to steer the system through unknown dynamics and collect the most informative states. While GPSSMs provide uncertainty estimates that guide exploration, they share computational scalability limitations with Kriging approaches and may be impractical for real-time testing applications.

2.3. Neural ordinary differential equations and integration with active learning

The aforementioned studies either rely on model uncertainty for data sampling, which can be computationally expensive and may not

scale well with large datasets, or assume scenarios in which data is already available, rather than providing a systematic framework for generating informative training data from scratch in experimental settings. In this work, ANNs are favored for their versatility, computational efficiency, and effectiveness in capturing nonlinearities, even though, unlike Kriging and GPSSMs, they do not provide direct access to uncertainty estimates. This trade-off is acceptable in the context where the focus is on practical implementation in testing environments with limited computational resources. To address these limitations, an active learning framework based on ANNs is proposed, using Neural ODEs as the modeling core due to their versatility and strong capability to represent nonlinearities. Neural ODEs are a class of ANNs particularly well-suited for modeling dynamic systems, as they embed the ODE framework required for such modeling based on time series. They offer continuous-time representations and can be trained end-to-end using standard backpropagation techniques, making them suitable for experimental applications. Bradley et al. (2024) emphasized that Neural ODEs require informative data that capture key system characteristics, which further motivates their integration with active learning. Specifically, Neural ODEs may fail to generalize when training data does not adequately span the system's dynamic range or when critical behavioral regimes are undersampled. This highlights the importance of integrating Neural ODEs with active learning strategies, which can guide data acquisition to ensure that key system dynamics are adequately captured while maintaining computational efficiency.

Concepts similar to the goals of the present study have been applied in automated protein design using self-driving laboratories (Rapp et al., 2024), where the integration of machine learning models with automated laboratory environments enabled systematic exploration of design spaces. While the experimental automation principles are transferable, the application domains differ significantly in terms of system dynamics, time scales, and measurement modalities. The idea behind the methodology proposed in this paper was conceptualized in our preliminary study (Milicevic and Altay, 2024), where simple feed-forward neural networks (FNNs) were used for learning force-state mapping from data. However, this approach requires the measurement of the restoring force and was validated only numerically, limiting its practical applicability to testing scenarios where force measurements may be difficult or expensive to obtain.

3. Methodology

TMDs consist of an oscillatory mass coupled to a structure and tuned to its natural frequency to generate optimal vibration-restoring forces. Most of these devices, such as the TLCD, rely on nonlinear energy dissipation mechanisms, making accurate modeling challenging with traditional analytical methods. This study focuses particularly on rate-dependent nonlinearities with weak memory effects, which are typical of TLCDs and arise from fluid viscosity and flow effects. It is assumed that TMDs can generally be represented by a single-degree-of-freedom (SDOF) system, as most are designed to operate in this manner or can be reduced to an equivalent SDOF system. Apart from its operational range, no pre-existing data or prior knowledge regarding the device's dynamic characteristics are assumed. Consequently, response data must be generated from scratch through laboratory experiments. Furthermore, the study aims to achieve the following two main objectives:

1. To autonomously collect response data containing representative information about the device's characteristics within its designated operational range, enabling the generation of an ANN model.
2. To minimize the number of experiments required to collect the aforementioned dataset.

An important component of the methodology is the use of Neural ODEs to model the control device, specifically its state response, as detailed in Section 3.1. Training this model, $\mathcal{M}$, in supervised learning mode requires fully labeled response data within the device's designated operational range. Each training dataset generated from the ith experiment is denoted as $D_i = \{\ddot{\mathbf{x}}_\mathbf{i}, \dot{\mathbf{u}}_\mathbf{i}, \mathbf{u}_\mathbf{i}\}$, containing the excitation signal, such as the acceleration $\ddot{\mathbf{x}}_\mathbf{i}$ of the shaking table, along with the velocity $\dot{\mathbf{u}}_\mathbf{i}$ and the displacement $\mathbf{u}_\mathbf{i}$ responses of the device. Supplying $\mathcal{M}$ with excessive data would require numerous experiments and could result in non-diverse datasets, such as disproportionately large amounts covering only specific behaviors, potentially leading to suboptimal training. In contrast, a very small dataset will lack the necessary information.

To circumvent these challenges, we propose an active learning framework that autonomously identifies an optimal training set $\mathcal{D}_{opt}$, encapsulating sufficient information to efficiently train the model and enable generalization. We define the vector $\boldsymbol{\psi}_i$ to represent the experimental conditions that can be controlled and optimized to generate $\mathcal{D}_{opt}$. These experimental parameters may include excitation signals, ambient temperature, or other factors, including those specific to the specimen. Within a predefined sampling space, as detailed in Section 3.2, a single set of parameters $\boldsymbol{\psi}_{opt}$ is searched for, such that conducting the experiment with this set generates the optimal training dataset $\mathcal{D}_{opt}$.

After training with $\mathcal{D}_{opt}$, the model $\mathcal{M}$ is evaluated on a separate reference dataset $\mathcal{D}_R$. The model $\mathcal{M}$ should then achieve a modeling accuracy within an error tolerance of ϵ. Classical performance tests for vibration control devices are typically conducted based on a predefined set of loading conditions dictated by application requirements and industry codes. In our approach, the data obtained with these loading conditions is used as the reference dataset $\mathcal{D}_R$ to evaluate the model accuracy. The outline of the proposed framework is shown in Fig. 1 with the following steps:

- **Step 1:** Sample an initial set of experimental parameters $\boldsymbol{\psi}_i = \boldsymbol{\psi}_1 \in \mathbb{R}^M$, where M represents the sampling space dimension. This can be done either based on prior experience or through random sampling.
- **Step 2:** Conduct performance testing using the parameters $\boldsymbol{\psi}_R$, as specified by the requirements of the planned application or governing standards, to obtain the reference dataset $\mathcal{D}_R = \{\ddot{\mathbf{x}}_\mathbf{R}, \dot{\mathbf{u}}_\mathbf{R}, \mathbf{u}_\mathbf{R}\}$. This step is performed only once. Conduct another experiment with the configuration $\boldsymbol{\psi}_i$ to obtain the training dataset $\mathcal{D}_i = \{\ddot{\mathbf{x}}_\mathbf{i}, \dot{\mathbf{u}}_\mathbf{i}, \mathbf{u}_\mathbf{i}\}$.
- **Step 3:** Use the obtained training set $\mathcal{D}_i$ to train the model $\mathcal{M}$ in supervised training mode. During training, Neural ODE employ a sliding temporal window approach where overlapping segments are randomly sampled from the time-series, utilizing all available data and validation is performed on new initial conditions. Once the model is trained, obtain the model error $\mathcal{L}_R\colon \mathbb{R}^M \rightarrow \mathbb{R}$ by evaluating the accuracy of the trained model on the reference set $\mathcal{D}_R$.
- **Step 4:** If any of the convergence criteria are met, such as the model error meeting the given tolerance ϵ, the process is terminated and the optimal training dataset is identified as $\mathcal{D}_{opt} = \mathcal{D}_i$. Otherwise, proceed to the next iteration, and sample the experimental parameters $\boldsymbol{\psi}_i$ with $i = i + 1$ according to the optimization algorithm (Section 3.2) to obtain a new training dataset.
  Repeat **Step 2** through **Step 4** until the convergence criteria are met. Once the criteria are satisfied, optimal training dataset is identified, and the model $\mathcal{M}$ is achieved.

This is also shown in more detail in Algorithm 1. It should be noted that the framework assumes that the chosen model class appropriately matches the complexity of the system, minimizing bias due to model

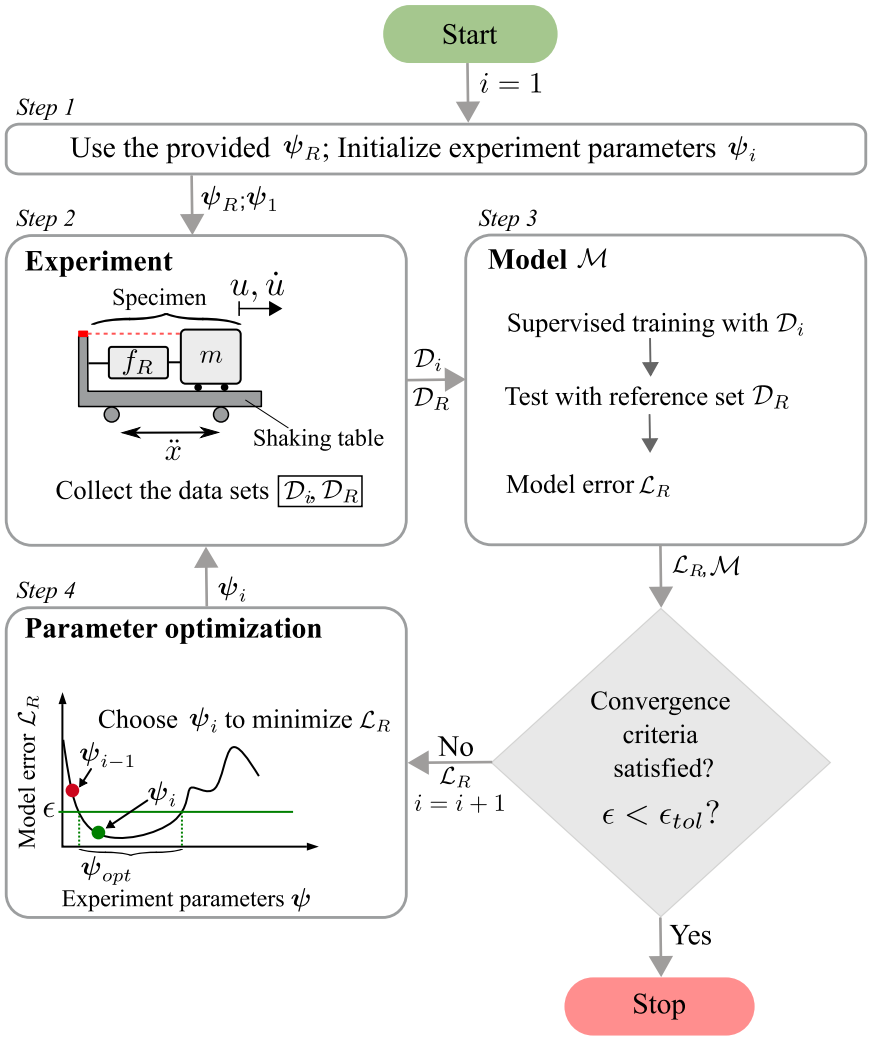

Fig. 1. Workflow of the proposed active learning framework. Here, ψ represents the parameters defining the experiment configuration, $\mathcal{M}$ is the target data-driven model of the control device, and $\mathcal{D}$ represents the dataset containing device response collected from experiments. $\mathcal{L}$ denotes the model error. The experimental setup shows the specimen's restoring force, f_R, and the auxiliary mass, m, which are modeled using data obtained from the shaking table.

capacity. By enhancing data quality, the framework aims to reduce bias resulting from data limitations. Even with a highly capable model, missing essential information in the data prevents ANN models from fully learning the underlying system dynamics. By incorporating data with relevant features, models are enabled to capture these relationships more effectively. Additionally, an ensemble of networks is employed to introduce diversity among the models. Each network has the same architecture but differs in initial weights and biases. This diversity reduces variance, as fluctuations due to initialization are smoothed across the ensemble. Readers interested in ensemble deep learning techniques are referred to the review paper (Ganaie et al., 2022).

3.1. Extended neural ordinary differential equations

Neural ODEs are a class of ANNs particularly well-suited for modeling dynamic systems, as they naturally embed the ordinary differential equation (ODE) framework required for such modeling based on time series. Classical Neural ODEs (Chen et al., 2018) are designed to model continuous dynamical systems by parameterizing their ODEs as:

$$\dot{\mathbf{h}} = f_\theta(\mathbf{h}, t, \boldsymbol{\theta}), \qquad \mathbf{h}(t_o) = \mathbf{h}_0, \tag{1}$$

where $\dot{\mathbf{h}} = \frac{d\mathbf{h}}{dt} \in \mathbb{R}^{2n}$ denotes the time derivative of the state vector, which is represented by $\mathbf{h} = [\mathbf{u}(t)\ \dot{\mathbf{u}}(t)]^\top$ with $\mathbf{u}(t) \in \mathbb{R}^n$ being the displacement and $\dot{\mathbf{u}}(t) \in \mathbb{R}^n$ the velocity, $t \in [t_0, T]$ is the time variable, t_0 denotes the initial time, $\mathbf{h}_0$ denotes initial conditions, and $\boldsymbol{\theta} \in \mathbb{R}^p$ represents the trainable parameters of the neural network $f_\theta : \mathbb{R}^{2n} \times \mathbb{R} \times \mathbb{R}^p \to \mathbb{R}^{2n}$. The parameters $\boldsymbol{\theta}$ consist of weights $\mathbf{W}$ and biases $\mathbf{b}$ of the FNN, and the output of each layer is computed as a weighted sum of inputs from the previous layer, activated with an activation function σ such that:

$$\mathbf{h}^l = \sigma^l(\mathbf{W}^l \mathbf{h}^{l-1} + \mathbf{b}^l), \tag{2}$$

where $\mathbf{h}^l \in \mathbb{R}^{n_l}$ denotes the hidden state at layer l, $\mathbf{W}^l \in \mathbb{R}^{n_l \times n_{l-1}}$ is the weight matrix, $\mathbf{b}^l \in \mathbb{R}^{n_l}$ is the bias vector, and $\sigma^l(\cdot)$ is the activation function of the lth layer. Using numerical integration methods, such as the Runge–Kutta algorithm, the state $\mathbf{h}$ at time t can be predicted from f_θ. In supervised training, the predicted solution is compared to the true solution through the loss function:

$$\mathcal{L}_N(\mathbf{h}(t)) = \mathcal{L}_N\left(\text{ODESolve}\left(\mathbf{h}(t_0), f_\theta, t_0, t, \boldsymbol{\theta}\right)\right), \tag{3}$$

where $\mathcal{L}_N : \mathbb{R}^{2n} \to \mathbb{R}$ is the loss function computed over N training samples, and ODESolve$(\cdot)$ denotes a numerical ODE solver that integrates the system from t_0 to t given initial state $\mathbf{h}(t_0)$ and dynamics f_θ. The loss is minimized during training by optimizing the parameters $\boldsymbol{\theta}$. This formulation assumes all system parameters can be learned implicitly through the network weights. However, while classical Neural ODEs are efficient for solving initial value problems, our framework requires

Algorithm 1 Active learning framework for tuned mass damper modeling

Require: Parameters $\boldsymbol{\psi}_R$, error tolerance ϵ, sampling space dimension M, ensemble size N_{ens}
Ensure: Optimal training dataset $\mathcal{D}_{opt}$ and trained model ensemble $\{\mathcal{M}_1, \dots, \mathcal{M}_{N_{ens}}\}$
Sample initial experimental parameters $\boldsymbol{\psi}_1 \in \mathbb{R}^M$ (randomly or based on prior knowledge)
Set iteration counter $i \leftarrow 1$
Set convergence flag `converged` $\leftarrow$ `False`
Conduct performance test with parameters $\boldsymbol{\psi}_R$
Obtain reference dataset $\mathcal{D}_R = \{\ddot{\mathbf{x}}_R, \dot{\mathbf{u}}_R, \mathbf{u}_R\}$
while `converged` = `False` **do**
  Conduct experiment with parameters $\boldsymbol{\psi}_i$
  Preprocess $\mathcal{D}_i$ (filtering, differentiation, normalization)
  Obtain training dataset $\mathcal{D}_i = \{\ddot{\mathbf{x}}_i, \dot{\mathbf{u}}_i, \mathbf{u}_i\}$
  for $j = 1$ to N_{ens} **do**
    Initialize Neural ODE model $\mathcal{M}_j$ with random weights
    Split $\mathcal{D}_i$ into training and validation subsets
    Train $\mathcal{M}_j$ on training subset with early stopping on validation subset
  end for
  Compute ensemble prediction on reference set: $\hat{\mathbf{u}}_R = \mathcal{M}_j(\mathcal{D}_R)$
  Compute model error: $\mathcal{L}_R(\boldsymbol{\psi}_i) = \text{Error}(\hat{\mathbf{u}}_R, \mathbf{u}_R)$
  if $\mathcal{L}_R(\boldsymbol{\psi}_i) \leq \epsilon$ **or** other convergence criteria met **then**
    $\mathcal{D}_{opt} \leftarrow \mathcal{D}_i$
    `converged` $\leftarrow$ `True`
  else
    $\boldsymbol{\psi}_{i+1} \leftarrow \text{PatternSearch}(\mathcal{L}_R, \boldsymbol{\psi}_i)$
    $i \leftarrow i + 1$
  end if
end while
return $\mathcal{D}_{opt}$, $\mathcal{M}_{opt}$

the modeling of forced vibrations. This is typically achieved by incorporating physical principles into Neural ODEs, as demonstrated in Lai et al. (2021). Furthermore, our approach assumes no prior knowledge about the vibration control device being modeled. To better illustrate that standard Neural ODEs cannot learn in case of forced dynamical systems, the state-space representation can be written as:

$$\dot{\mathbf{h}}(t) = \mathbf{A}\mathbf{h}(t) + \mathbf{B}F(t) \tag{4}$$

where $\mathbf{A}$ is the state matrix, $\mathbf{B}$ is the input matrix, and $F(t)$ is the external forcing. For structural systems with n degrees of freedom, the input matrix contains mass-dependent terms. For example, in a linear SDOF system:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}, \tag{5}$$

where m, k and c denote the mass, stiffness, and viscous damping coefficient, respectively.

Hence, when mass parameters are unknown, the standard Neural ODE faces an identifiability issue. The network cannot uniquely determine both the system dynamics ($\mathbf{A}$) and the input scaling ($\mathbf{B}$) simultaneously, as these are coupled through the unknown mass. To address this, the modification presented in this paper explicitly parameterizes the input scaling and extends the Neural ODE formulation in Eq. (1) as follows:

$$\dot{\mathbf{h}} = f_\theta(\mathbf{h}, t, \theta) + \theta_m \mathbf{B}F(t), \tag{6}$$

where $F(t) \in \mathbb{R}^m$ is the external force excitation, $\theta_m \in \mathbb{R}$ is a scalar parameter that accounts for unknown modal mass, and $\mathbf{B} \in \mathbb{R}^{2n \times m}$ is the input matrix. Corresponding to an SDOF system, $\theta_m = -m^{-1}$ and $\mathbf{B} = [0 \quad 1]^\top$ are used. Note that θ_m is not explicitly used in the FNN. However, it is optimized simultaneously with other parameters, $\boldsymbol{\theta}$, using the same loss function, ultimately enabling the identification of the modal mass. Hence, this extension allows:

1. The neural network f_θ to focus on learning the intrinsic system dynamics.
2. The parameter θ_m to capture the mass-dependent input scaling.
3. Unique identification of both components.

Fig. 2 shows the training procedure and the computation of the training loss in Eq. (1). The training is done iteratively in the following manner:

1. the dataset $\mathcal{D}$ contains measured system states $\mathbf{h}(t)$ and input forces $F(t)$ at discrete time points $t = t_1, \dots, t_n$. For each training iteration, an initial condition $\mathbf{h}(t_{s,i}), F(t_{s,i})$ is randomly selected from the training set and used as input for the Neural ODE.
2. using the initial condition from Step 1, numerical integration is performed to predict the solution at time t_p for iteration i. Note that the prediction horizon is a hyperparameter of the Neural ODE and can be tuned. The integration incorporates the FNN f_θ, which models the system's restoring acceleration, and the free parameter θ_m, which models the system's mass and scales the input force.
3. after solving the ODE in Step 2, the output $\hat{\mathbf{h}}(t_{p,i})$, as well as the trained parameter θ_m are obtained.
4. The predicted response $\hat{\mathbf{h}}(t_{p,i})$ from Step 3 is compared to the true value $\mathbf{h}(t_{p,i})$, and the loss $\mathcal{L}_N$ is calculated.
5. the loss from Step 4 is used to update the parameters $\boldsymbol{\theta}$ and θ_m via backpropagation.
6. a new iteration is initiated with fresh initial conditions selected from the dataset.

3.2. Optimization algorithm

To find an optimal training set $\mathcal{D}_{opt}$ while keeping the number of experiments as small as possible, we formulate an optimization problem in which the experimental parameters $\boldsymbol{\psi}_i$ are adjusted to minimize the model error $\mathcal{L}_R(\boldsymbol{\psi}_i)$. Since certain samples of the experimental parameters will lead to more informative datasets, the error function will have at least one global minimum with possible local minima. However, the model error comprises both the error due to data quality — our primary focus — and errors stemming from variance components and the suitability of the selected architecture. Consequently, the model error is stochastic, necessitating a derivative-free optimization method. Most optimization algorithms designed for stochastic problems are compatible with our framework. For example, particle swarm optimization (Guo et al., 2025) could also be utilized, but it requires many samples. However, since our goal is to minimize the number of experiments, we prioritize optimization algorithms that require fewer function evaluations (experiments). In this regard, the pattern search methods are particularly suitable, as they are derivative-free (Torczon, 1997). For at least a moderate number of variables, they can handle non-smooth problems with fewer function evaluations compared to other methods, such as genetic algorithms. With pattern search, experiments are executed sequentially in our approach, with new experimental parameters ${}^k\boldsymbol{\psi}_i^e$ sampled at each iteration using a *mesh*:

$${}^k\boldsymbol{\psi}_i^e = \boldsymbol{\psi}_i + \Delta \cdot {}^k\boldsymbol{\nu}, \tag{7}$$

where k denotes the number of mesh points, corresponding to the number of *patterns* ${}^k\boldsymbol{\nu}$ chosen based on the sampling space dimension M. Furthermore, Δ is the step length equivalent to the Euclidean distance between the exploratory steps ${}^k\boldsymbol{\psi}_i^e$ and the mesh center $\boldsymbol{\psi}_i$

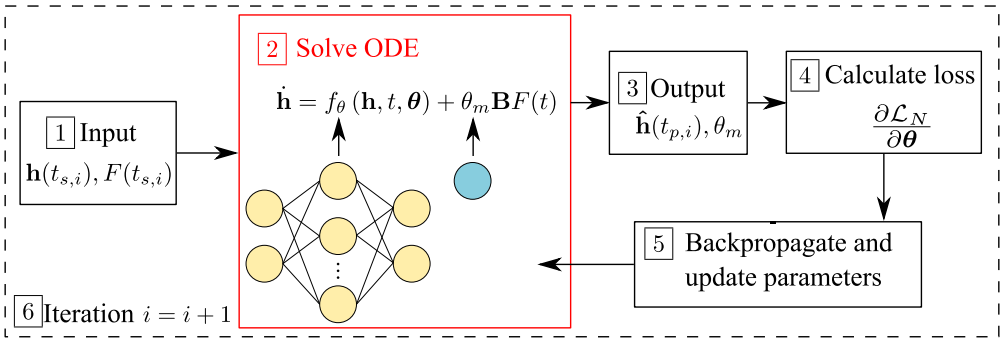

Fig. 2. Expanded Neural ODE architecture and learning process.

representing the best current solution. If any of the exploratory steps produces a better solution such that $\mathcal{L}_R({}^k\boldsymbol{\psi}_i^e) < \mathcal{L}_R(\boldsymbol{\psi}_i)$, it becomes the new mesh center $\boldsymbol{\psi}_{i+k} = {}^k\boldsymbol{\psi}_i^e$. The execution can be done such that all k exploratory steps are executed first and the best solution is then evaluated, or using a greedy approach, where the mesh center is moved immediately to the first better solution. If the exploratory step is successful, meaning a better solution than the current one is found, the mesh is expanded. If no improvement is achieved, the mesh is contracted, with the current mesh center remaining unchanged $\boldsymbol{\psi}_{i+k} = \boldsymbol{\psi}_i$. This approach enables the algorithm to balance exploration and exploitation within the sample space. Contraction continues until a predefined minimum mesh size Δ_{tol} is reached, at which point the algorithm terminates. Additional stopping criteria include cases where the model error tolerance is satisfied such that $\mathcal{L}_R(\boldsymbol{\psi}_i) < \epsilon$ or a predefined maximum number of experiments is reached. The stopping criteria are application-driven and are defined by the user.

4. Experimental validation on a tuned liquid column damper

A TLCD is an auxiliary mass control device consisting of a U-shaped tube filled with Newtonian liquid (Altay and Klinkel, 2018). It dissipates energy through the movement of the liquid and head loss resulting from directional changes and restrictions within the tube. Due to its geometric versatility and ability to achieve low frequencies, various TLCD layouts have been developed for civil engineering structures, such as skyscrapers and wind turbines. TLCDs are used both as standalone systems or in combination with other vibration control systems, such as base isolation (Furtmüller et al., 2019).

In this section, we experimentally validate the proposed framework using a TLCD prototype. We demonstrate how the optimal training dataset is autonomously identified to develop a Neural ODE model while minimizing the number of experiments. Additionally, we compare the performance of the proposed framework with several standard unsupervised sampling methods. Furthermore, the data used for the presented results come from the experiments conducted in the laboratory and the simulations and investigations were performed in MATLAB R2023a. The neural network architecture was implemented using the *Deep Learning Toolbox*. The pattern search optimization was performed using the *Global Optimization Toolbox* with the *patternsearch* function.

The study is structured as follows. Sections 4.1 and 4.2 introduce the experimental setup, parameters, and sampling space. Sections 4.3 and 4.4 describe the optimization algorithm parameters and the Neural ODE configuration and training procedure. Section 4.5 presents the investigations and results, including the error surface (Section 4.5.1), a comparison with unsupervised methods (Section 4.5.2), Neural ODE training results and TLCD modal mass identification (Section 4.5.3), model robustness (Section 4.5.4), architectural modifications (Section 4.5.5), optimization parameter studies (Section 4.5.6), and a comparison with a genetic algorithm (Section 4.5.7).

4.1. Experimental setup

The TLCD prototype used in this study is an omnidirectional TLCD mounted on a shaking table (Fig. 3, ①). Omnidirectional TLCDs are capable of controlling vibrations in multiple lateral directions. As described in Mehrkian and Altay (2020), their dynamic response can be represented by an SDOF system, independent of the excitation direction. The prototype features four L-arms, each divided into multiple cells ②. Liquid oscillation is measured using an ultrasonic level sensor ③ with a sampling rate of 100 Hz. Each cell can be opened or closed by actuators ④ to achieve different natural frequencies, providing the damper with semi-active vibration control capabilities, as described in Mehrkian and Altay (2025). For brevity, the proposed framework is demonstrated using a single configuration, and the TLCD is excited uniaxially. An inlet/outlet pipe ⑤ is used to fill the damper. The fluid motion in a TLCD is described by a streamline. The idle liquid height is $L_V = 125$ mm, and the horizontal distance between the column axes is $L_H = 520$ mm. The cells c_1 and c_2 are 70 mm wide, while cell c_3 is 50 mm wide. All cells have a length of $d_3 = 50$ mm.

To illustrate the complexity of the system expected to be modeled, the analytical representation of the TLCD is presented, which is derived using the unsteady Bernoulli equation:

$$\ddot{u} + \delta|\dot{u}|\dot{u} + \omega_D^2 u = \gamma_1 \ddot{x} \tag{8}$$

where $u(t) \in \mathbb{R}$ denotes the liquid displacement, $\dot{u} = \frac{du}{dt}$ is the liquid velocity, $\ddot{u} = \frac{d^2u}{dt^2}$ is the liquid acceleration, $\ddot{x}(t) \in \mathbb{R}$ is the excitation acceleration applied to the damper, $\delta > 0$ is the head loss coefficient representing nonlinear damping effects, $\omega_D > 0$ is the natural circular frequency of the damper, and $\gamma_1 \in \mathbb{R}$ is a dimensionless geometric factor relating the excitation to the liquid motion. This equation of motion of the TLCD is assumed to be unknown in the proposed framework. The damper is characterized by its natural circular frequency, ω_D, which can be analytically modeled with high accuracy (Mehrkian and Altay, 2020). Another critical parameter is the nonlinear damping term $\delta|\dot{u}|\dot{u}$, where the head loss coefficient δ accounts for flow losses due to pipe friction and turbulence. While some analytical formulations exist, their accuracy is limited for complex TLCD layouts, as investigated in Mehrkian and Altay (2025). Estimating the geometric factor γ_1 is equally challenging. During the derivation of the equation of motion, this factor is introduced to consolidate the damper's geometric parameters. The geometric factor γ_1 is defined as:

$$\gamma_1 = \frac{L_H}{2L_V + \frac{A_V}{A_H} L_H}, \tag{9}$$

where A_V/A_H represents the ratio of the cross-section areas of the vertical and horizontal parts of the damper. Since L_H and L_V approximate the liquid flow direction, they are particularly challenging to determine for complex TLCD layouts, which impacts the accuracy of γ_1. However, ensuring the accuracy of this factor is crucial, as it corresponds to the modal mass of the damper, as discussed in Altay and Klinkel (2018),

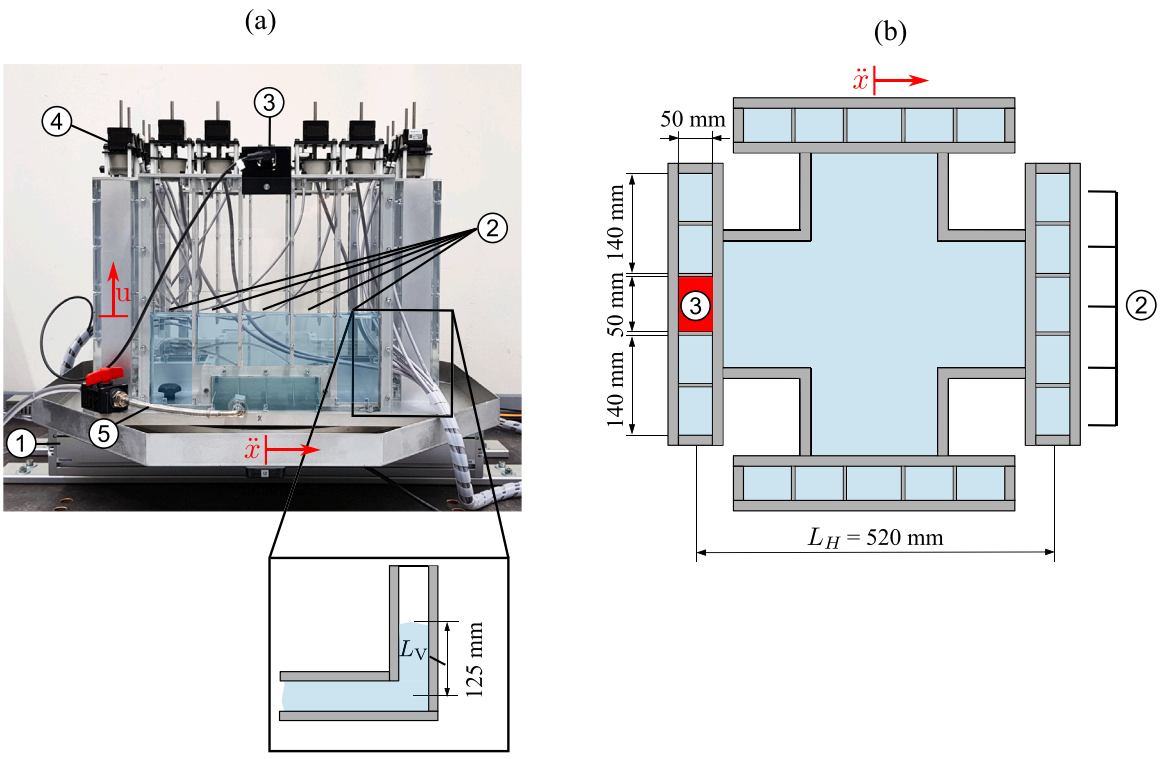

Fig. 3. Experimental setup of the TLCD prototype with four L-arms connected to a uniaxial shaking table ①, shown in the front view (a) and plan view (c), along with corresponding sketches (b) and (d). Each L-arm is divided into five cells ②. Liquid motion is measured using ultrasonic sensors ③, while actuators allow the cells to be opened or closed ④. Liquid can be added or removed through an inlet/outlet pipe ⑤.

and is essential for designing structures equipped with TLCDs. In its expanded version, the Neural ODE can directly estimate the geometric factor $\hat{\gamma}_1$ to circumvent this challenge, cf. Section 4.4.

4.2. Experimental parameters and sampling space

As shown in Fig. 4, the excitation signal applied to the shaking table is parameterized while keeping other experimental conditions constant. Thus, simple harmonic excitation is used to generate training datasets D_i despite the potential superiority of other signals for model identification. By retaining the transient response in the signal, it is assumed that the signal is sufficiently powerful to extract the necessary information. The goal is not to design the optimum excitation signal but rather to sample the most informative response data. Using a simple harmonic function allows for a two-dimensional parameterization: the excitation amplitude x_0 and cyclic excitation frequency f_S. This approach enables visualization of the model error $\mathcal{L}_R(\boldsymbol{\psi})$, which would not be feasible with higher-dimensional parameterization.

The sampling space of the experimental parameters, the excitation amplitudes and cyclic excitation frequencies, is defined based on the damper's operational range, which is determined by the maximum possible liquid displacement. The measured damper response (liquid level) and shaking table motion (base acceleration) are preprocessed. First, the raw measurement signals are filtered to remove high-frequency noise. Next, for training the Neural ODEs, the excitation signal is translated into an acceleration $\ddot{x}$ time history. Offsets and high-frequency noise are removed from the measured liquid displacement u, and the corresponding velocity $\dot{u}$ is obtained through numerical derivation. Finally, the signals are then truncated to 20 s to obtain the same number of measurement points for each dataset. The response signals u and $\dot{u}$ serve as direct inputs to the Neural ODEs, while the excitation signal $\ddot{x}$ is added to the network output before integration.

Fig. 5 displays the sampling space for the experimental parameters. To gain a comprehensive understanding of the results, the experimental parameters are initially sampled uniformly before applying the active learning framework. These sampling points are represented as dots in the figure. The displacement amplitudes are sampled in the range of 5–40 mm, with increments of 5 mm. The excitation frequencies are sampled in the range of 0.40–0.96 Hz, with increments of 0.08 Hz, and additionally at 1.00 Hz. With active learning, the optimization algorithm will be restricted to selecting only from these points. If a sample is requested that does not correspond to these points, the point closest to the requested one will be used. While this restriction limits the effectiveness of the proposed framework, it allows for a more accurate comparison with unsupervised sampling methods by utilizing the same set of available points. An additional challenge arises from the indirect nature of the sampling process, as mentioned in the introduction. Although the goal is to obtain the training data, the testing-integrated framework prevents direct sampling. Instead, the excitation signal must be sampled first and applied to the testing system to generate the training data. Direct sampling methods are therefore not applicable. Furthermore, as indicated by the background color in Fig. 5, the acceleration amplitude of the shaking table increases quadratically with increasing frequency. This nonlinearity introduces another challenge in identifying optimal training data.

Fig. 6 displays the reference dataset D_R used to evaluate the model error. Fig. 6(a) shows the sine-sweep excitation signal. A sine sweep was used because it effectively extracts comprehensive information about the system and serves as a reliable measure of model accuracy and generalization capability. The signal has a constant acceleration amplitude of 0.4 ms^{-2} and spans an excitation frequency range of 0.40–1.00 Hz. With this excitation signal, the displacement response measurement shown in Fig. 6(b) was obtained. The displacement measurement was numerically differentiated to obtain the velocity response shown in Fig. 6(c). It is evident from Figs. 6(b) and 6(c) that measurement noise from the sensors and the shaking table is present in the datasets. Therefore, the dataset replicates a realistic scenario and poses an additional challenge for the trained model. The important parameters of the experiments are summarized in Table 1.

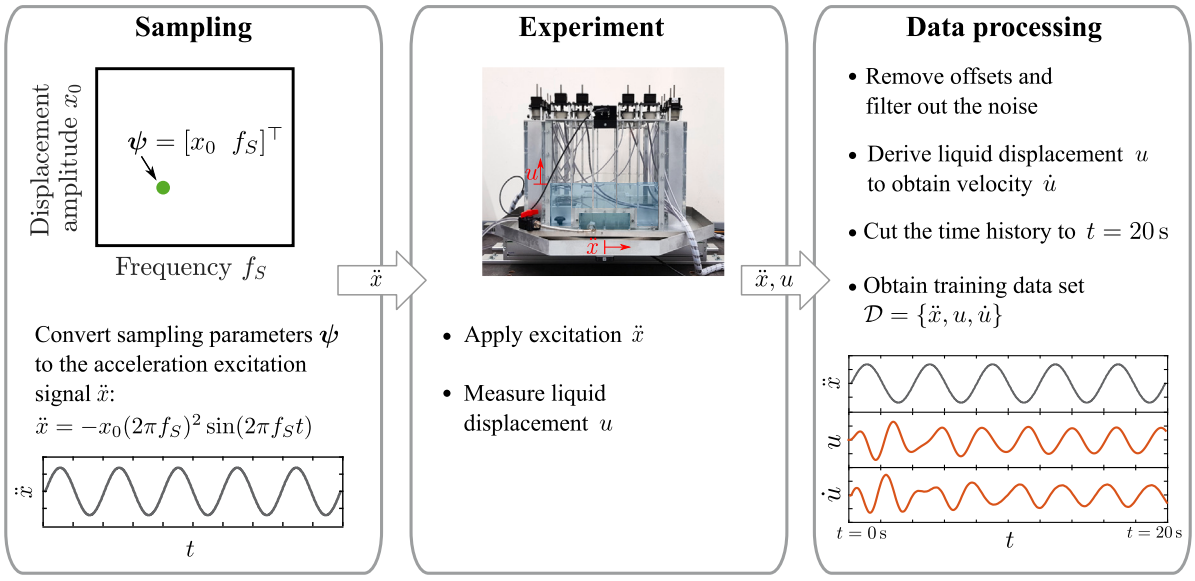

Fig. 4. Workflow illustrating the process of obtaining a training dataset from a sampling point.

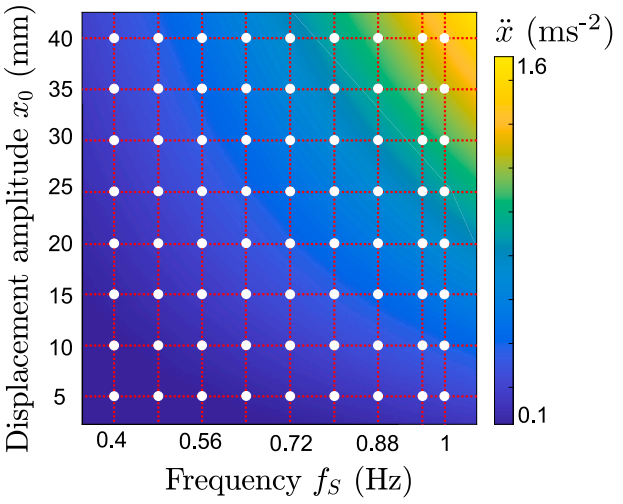

Fig. 5. Plot showing the relationship between displacement amplitude x_0, cyclic excitation frequency f_S, and acceleration amplitude $\ddot{x}_0 = -x_0(2\pi f_S)^2$.

Table 1
Important parameters of the experiments.

	Parameter	Value
Damper: Tuned Liquid column damper	Liquid level (L_V)	125 mm
	Column distance (L_H)	520 mm
	Cross section (vertical) (A_V)	16500 mm^2
	Cross section (horizontal) (A_H)	8500 mm^2
Training signal	Signal type	single-sine excitations
	Amplitude range (x_0)	5–40 mm
	Frequency range (f_S)	0.40–1.00 Hz
Reference signal	Signal type	Sine-sweep excitation
	Acceleration amplitude ($\ddot{x}_0$)	0.4 m/s^2
	Frequency range (f_S)	0.40–1.00 Hz

4.3. Pattern search parameters

The experimental parameters, $\boldsymbol{\psi}_i = [x_{0_i} \quad f_{S_i}]^\top$, serve as inputs. The output is the model error $\mathcal{L}_R$. The error measure used is the mean absolute error (MAE). The first guess for the experimental parameters $\boldsymbol{\psi}_1$ is random, and each subsequent set of experimental parameters is determined by the optimization algorithm. The parameter space was scaled such that the boundaries are between 0 and 10 for all sampling parameters. The *minimal basis set* was used for the poll method, where the mesh points were determined by ${}^k\boldsymbol{\psi}_i^e = \boldsymbol{\psi}_i + \Delta \cdot {}^k\boldsymbol{v}$ with ${}^k\boldsymbol{v} = \{[1 \quad 0]^\top, [0 \quad 1]^\top, [-1 \quad -1]^\top\}$. To further minimize the number of experiments, a greedy approach was utilized by disabling *complete poll* option. In addition the *cash option* was enabled, which allows us to keep the evaluation record such that the same point is not evaluated twice. The mesh tolerance Δ was set corresponding to the distance measured between the two sampled parameters in the experimental campaign described in Section 4.2, such that when the mesh size becomes smaller than the distance between two points on the grid (shown in Fig. 5), the algorithm terminates. After each successful iteration, the mesh was expanded by a factor of 2, which is the default setting in MATLAB and was not further optimized.

Therefore, the convergence criteria were carefully selected to balance accuracy requirements with practical experimental constraints. The error threshold was set to 0.25 mean average error, which approximately translates to 0.1 normalized root mean square error, where normalization is performed using standard deviation. This corresponds to approximately 90% goodness of fit. This threshold was determined to be appropriate as it allows for both measurement noise and uncertainty as well as numerical processing and integration errors. The minimum allowed mesh size was set to 0.2, which corresponds to a distance in the measurement grid where performance does not vary significantly. The algorithm was allowed to run until either the minimum mesh size was reached or the minimum error threshold was satisfied, without restrictions on the maximum number of experiments.

4.4. Neural ordinary differential equation setup

Since the Neural ODE consists of an internal FNN that essentially learns a force-state map, and the anticipated nonlinearity has weak memory, the network architecture does not need to be deep, as discussed in Pei et al. (2022). Therefore, to model the TLCD, a Neural ODE consisting of a fully connected layer with 20 neurons and a *sigmoid* activation function is used. The architecture has not undergone further optimization but initial tests suggest that it is sufficiently large to account for the expected system complexity, while remaining compact enough to avoid introducing significant variance error.

A sliding-window training scheme is adopted that preserves temporal causality. The model is trained using contiguous, fixed-length

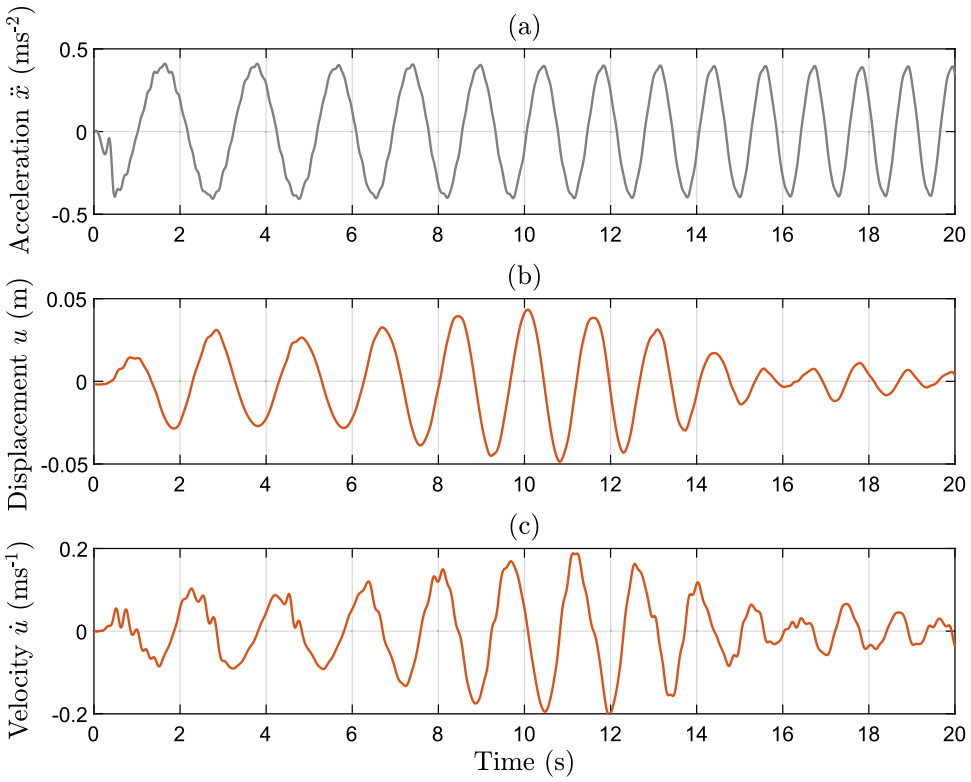

Fig. 6. Reference dataset obtained using a sine-sweep excitation. The excitation signal is shown in (a), and the damper displacement and velocity responses are shown in (b) and (c), respectively.

windows, each constructed by selecting random starting indices from the full time series. In this way, 100% of the signal contributes to training over the course of optimization (though not simultaneously), and no shuffling of time steps occurs, ensuring physically meaningful time dependencies. For validation, a held-out set of starting indices from the same signal (i.e. a segment-level split) is used at each iteration. The evaluation windows were also contiguous and did not reuse any of the training start positions, so the validation assesses generalization to unseen segment initial conditions rather than unseen time samples. The training parameters were selected based on preliminary tests to achieve a good balance of accuracy and stability. These intermediate results were omitted from the manuscript to maintain conciseness. The model was trained for 10^4 iterations with a batch size of 200. The Neural ODE time-length parameter N_t specifies the number of prediction steps in the Neural ODE during the training. To account for the noise present in the dataset it was set to 40. This value was found as optimal, as smaller values of N_t would lead to poor training and model accuracy, while larger values would increase the training time, but would not lead to better accuracy. The network parameters were initialized using Glorot initialization. The Adam (Adaptive Moment Estimation) optimization algorithm was employed with a learning rate of 0.01 and a gradient decay rate of 0.9 The training loss was obtained using the MAE as the error function. Furthermore, an ensemble of ten Neural ODEs was employed to reduce the training variance, each with identical architecture and training parameters but initialized with different random weights. After each training, all ensemble members were evaluated, and the member with the best performance is taken as the TLCD model. Finally, the modifications described in Section 3.1 are utilized to incorporate the excitation signal and the identification of the geometric ratio γ_1, where the Neural ODE prediction is obtained from $\theta_m = -\hat{\gamma}_1$.

4.5. Results and discussion

4.5.1. Error surface

To better demonstrate the performance of the proposed framework, the model error over the entire sampling space is visualized. We emphasize that the error surface is created solely for illustrative purposes and is not part of the methodology, nor is it assumed to be known. Here, the Neural ODE ensemble is trained sequentially with the response data corresponding to the sampling points. After each training, the obtained model is evaluated on the reference dataset to obtain the model error. By plotting each corresponding $\mathcal{L}_R(\boldsymbol{\psi})$ for each $\boldsymbol{\psi}$ in the sampling range, the error surface can be obtained and therefore the objective function can be visualized. Fig. 7(a) illustrates the error surface, where a dark blue color indicates a more informative training dataset, and red indicates a poor dataset. Each number inside the square on the graph shows the MAE of the model. Hence, the surface regions shown in blue correspond to smaller errors and are the regions of high model accuracy. As described in Section 4.4, the errors shown in Fig. 7(a) correspond to the best trained member of the Neural ODE ensemble. It is evident that the best performance is at $\boldsymbol{\psi} = [20\,\text{mm}\ \ 0.64\,\text{Hz}]^\top$. Knowing that the natural frequency of the system, evaluated in a separate study, is 0.64 Hz, this expectation is met. However, the minimum model error does not align with the maximum excitation amplitude, contrary to intuitive expectations, underlining the need for the optimization algorithm. This discrepancy arises due to the simultaneous identification of the geometric scaling factor γ_1 and the state matrix by the Neural ODE. As the excitation amplitude increases, the transient stage of the response signal becomes less prominent, leading to non-unique solutions and, consequently, a less accurate model. Fig. 7(b) illustrates the smoothed error surface, which was obtained through linear interpolation of the discrete results.

Fig. 8 compares the reference signals with the response predicted by the model trained on the most informative data sampled at $\boldsymbol{\psi} =$

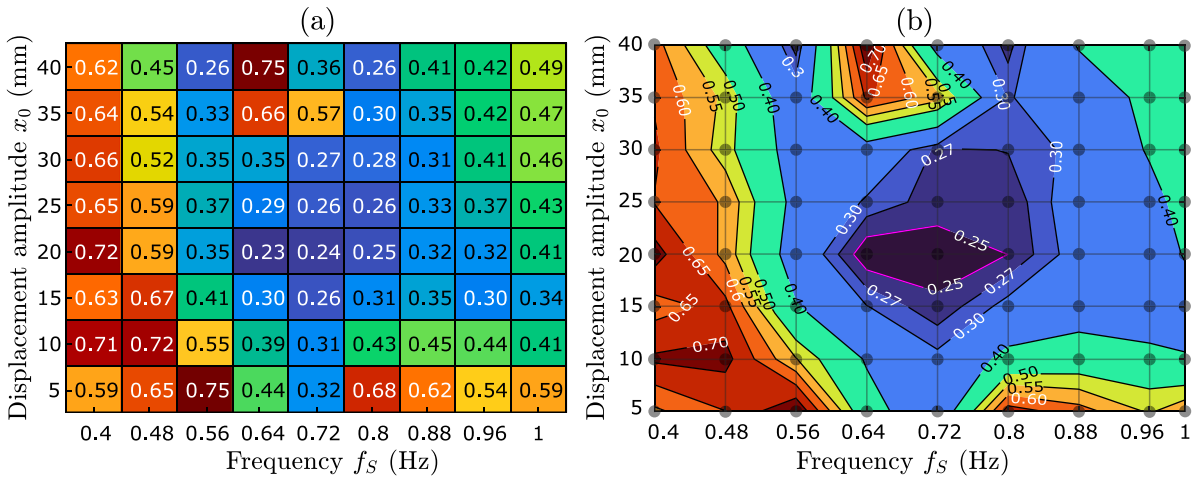

Fig. 7. (a) Error surface representing the information content of each sampling point. Squares indicate the training sets obtained using a sine excitation with the shown amplitude and frequency. The displayed value represent the mean absolute error (MAE) of the model when evaluated using a sine sweep, where red indicates high error. (b) Linear interpolation of (a) using the grid points, providing a smooth representation of the error surface.

$[20\,\text{cm}\ 0.64\,\text{Hz}]^\top$. As shown, the model accurately replicates the unseen reference signal. From the error surface (Fig. 7) and the depicted responses (Fig. 8), the following results can be interpreted:

- The error surface has a minimum, which justifies the assumption that there is a region in the sampling space that is more informative.
- The error surface is stochastic, hence the use of a stochastic optimization algorithm is justified.
- The optimal training dataset is attainable as shown in Fig. 8. This result confirms that the simple harmonic excitation is powerful enough to extract data from this system, such that the trained model achieves the desired accuracy on the reference set. Furthermore, the trained model can handle a significant levels of noise present in the training set, as shown in Fig. 8(c) where it can be seen that the model prediction is smooth.

4.5.2. Comparison to unsupervised sampling methods

Three different unsupervised sampling methods are used as benchmarks to investigate the performance of the proposed methodology:

- Uniform Random Sampling,
- Latin Hypercube Sampling (LHS),
- Sobol Sequence Sampling.

For the Uniform Random Sampling and LHS, the default settings of MATLAB were used. The Sobol set was generated using the option to skip the first 1000 points in the sequence and to take every 101^{st} point after the initial skip. Both of these settings are used to improve the quality of the obtained distribution by increasing the uniformity of the distribution of the sampled points.

For the unsupervised sampling methods, the number of sampled points is predetermined and the sampling point distribution in the sampling space depends on the method itself. Since the aforementioned unsupervised methods are quasi-random, the sampling point distribution they generate will be different each time they are applied. Comparison with unsupervised sampling methods is a common approach for evaluating active learning techniques, and it has been shown that active learning methods do not always outperform them (Yang and Loog, 2018). To obtain a statistically viable picture, the sampling methods were run 1000 times, each time sampling 5, 10, 15, 20 points. The success rate was determined by evaluating the model error and identifying the number of successfully trained models out of 1000 runs.

Table 2
Success rates of different sampling methods across varying number of experiments, evaluated over 1000 independent runs. The proposed active learning framework achieves 90% success rate with significantly fewer experiments than unsupervised methods.

	Number of experiments				
Method	10	20	30	40	50
Uniform Random Sampling	34.2%	54.9%	64.9%	75.3%	85.3%
Latin Hypercube Sampling	38.6%	65.7%	75.6%	85.7%	>90%
Sobol Sequence	37.6%	66.0%	82.0%	>90%	>90%
Active Learning (Proposed)	50.2%	>90%	>90%	>90%	>90%

For example, using random sampling with 20 points, the simulation is run 1000 times with different random distributions of 20 points, and $\mathcal{D}_{opt}$ was successfully identified 549 times, hence the success rate of 54.9%. The results are shown in Fig. 9(a) and summarized in Table 2 showing that the proposed active learning framework clearly outperforms all unsupervised sampling methods across all sample sizes tested. Here, the number of sampling points required by each method to achieve a success rate of at least 90% is also shown. For a fair comparison, the proposed active learning framework was initialized 1000 times with the *Halton Sequence Sampling*. This approach ensured a uniform distribution of the starting points and eliminated any bias that could result from the initial sampling. The number of sampling steps required to identify the optimal training dataset was counted for each sequence to obtain the success rate. After running all 1000 simulations, the histogram shown in Fig. 9(b) is obtained and shows the frequency of success of the proposed framework. The proposed methodology clearly outperforms the unsupervised sampling methods by reaching a success rate of 90.4% with only 20 experiments. In comparison, the best-performing unsupervised sampling methods, LHS and Sobol sequence, achieve with the same number of experiments a success rate of about 66% and require at least 35 to 45 experiments to reach the same success rate as the proposed approach.

Fig. 10 illustrates example cases of points placements generated by the unsupervised sampling methods and the proposed active learning framework on the smoothed error surface. Since the proposed approach requires in average 10 experiments (cf. Fig. 9(b)) to achieve the defined success rate, this case is depicted in the figure. It should be noted that the efficiency of unsupervised sampling methods is enhanced by the constrained sampling space. Expanding the sampling space would make it more challenging for these methods to effectively cover the area. Consequently, the benefits of the proposed method are expected to be even more pronounced in such scenarios.

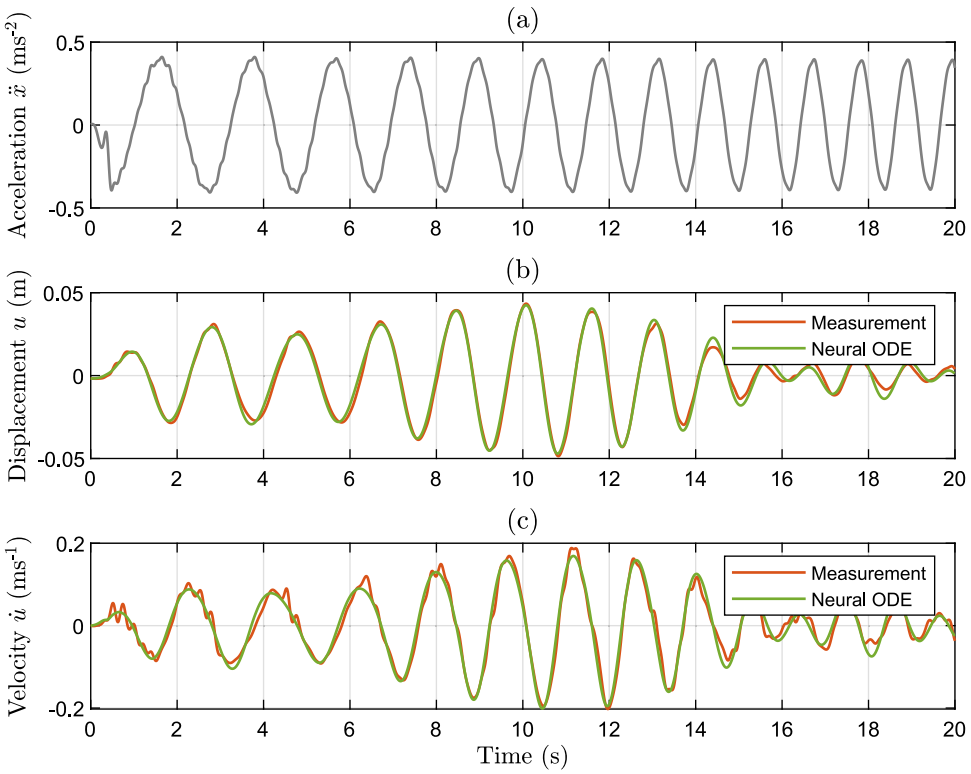

Fig. 8. Plots showing the reference dataset obtained using sine-sweep excitation shown in (a) with the damper displacement (b) and velocity (c) shown in green. Furthermore (b) and (c) show the performance of the Neural ODE when trained with the optimal training dataset obtained by the proposed framework.

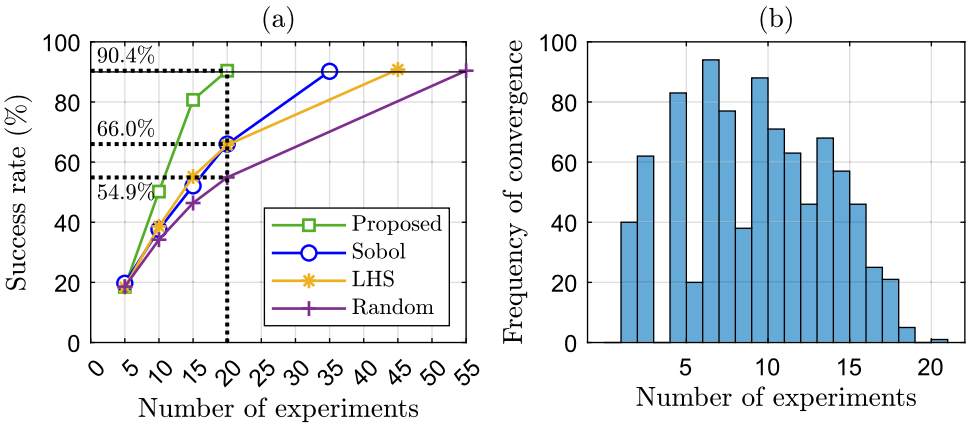

Fig. 9. (a) Success rate of the proposed active learning framework compared to unsupervised sampling methods. The plot shows that the proposed method achieves the success rate of 90% with only 20 samples, whereas other methods require at least 35 samples to achieve the same performance. (b) Histogram showing the frequency of convergence for the proposed framework, showing the number of experiments which was required to find the optimum training data set for each of 1000 runs.

4.5.3. Training of the neural ordinary differential equation

Table 3 summarizes the Neural ODE architecture, training parameters, and performance metrics obtained with the optimal training dataset. Fig. 11(a) shows the training loss obtained using the MAE as the error function for the optimal training dataset, indicating that the model is learning effectively from the data. The training starts with a rapid decline reaching a steady convergence after around 7000 iterations. Fig. 11(b) depicts the estimated θ_m, which provides the geometric factor as $\hat{\gamma}_1 = -\theta_m$, converging to 0.40. This result closely matches the theoretical value obtained according to analytical methods from (Mehrkian and Altay, 2020, 2025). Note that the identification of the geometric factor is enabled by the Neural ODE setup and would not be possible if a classical FNN were used, as the FNN would require a measurement of the restoring force. In the presented case, the Neural ODE model was able to identify a meaningful geometric factor, even when initialized with a physically impossible negative value demonstrating the robustness of the proposed approach.

Furthermore, the effectiveness of using an ensemble of Neural ODEs to eliminate the training variance is investigated. Fig. 12(a) presents the error surface observed when a single Neural ODE is utilized. This figure highlights the variance and potential instability in the training process when relying on a single model. Fig. 12(b) demonstrates the improvement in the error surface when the network ensemble is used, indicating a more stable outcome. The shown values are calculated by subtracting the error surface obtained using the ensemble (Fig. 7(a)) from the error surface obtained using a single Neural ODE (Fig. 12(a)). Using an ensemble, variance is significantly reduced, leading to more consistent and accurate parameter identification and model generation.

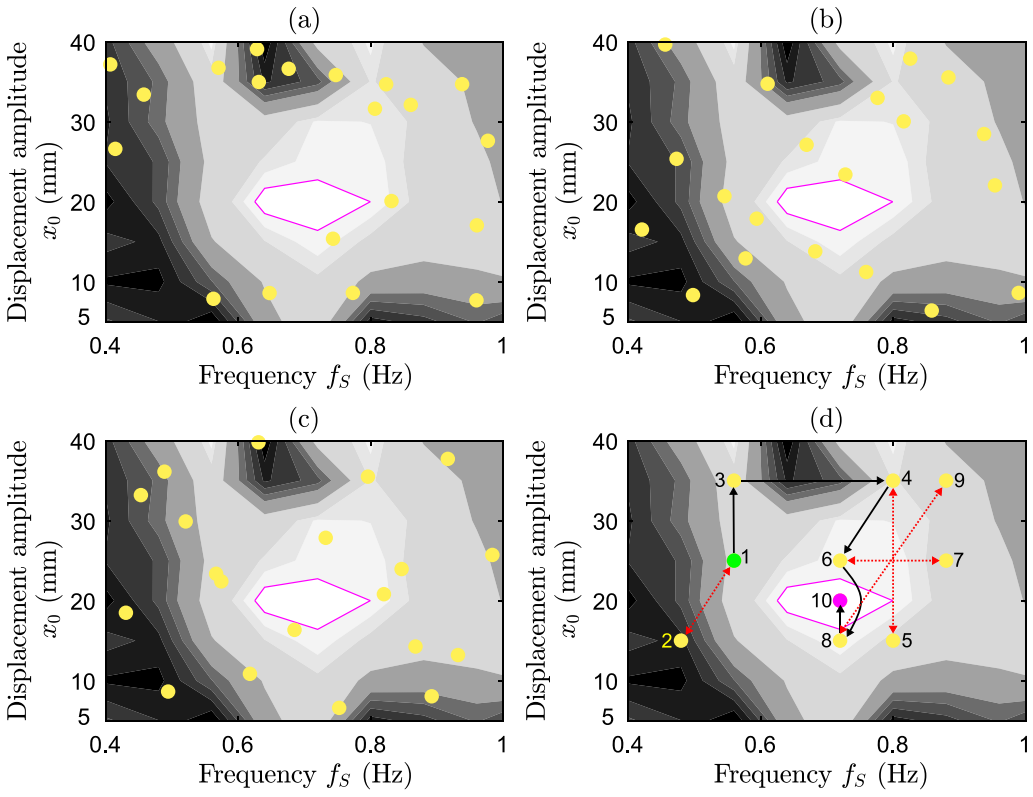

Fig. 10. Figures showing an instance of sampling point distribution using (a) Uniform Random Sampling, (b) LHS and (c) Sobol Sequence Sampling methods. Figure (d) shows an example of the typical convergence with the proposed active learning framework. The framework starts with an initial sampling point (green) and finds the optimal training dataset (magenta).

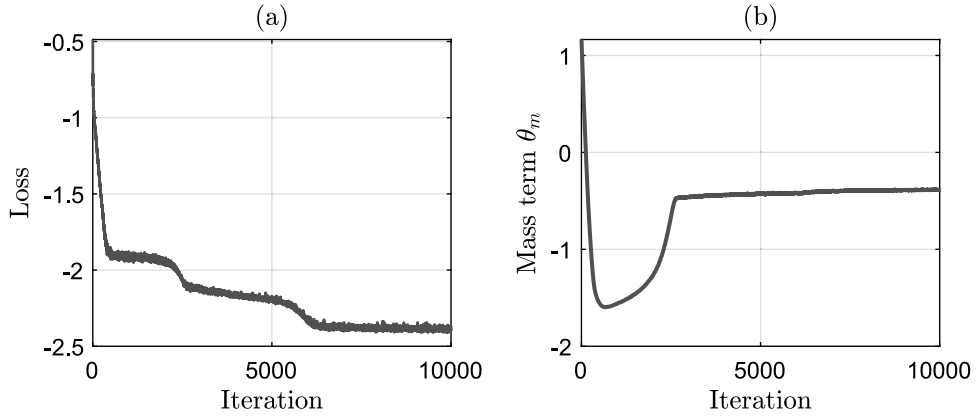

Fig. 11. (a) Training history of the Neural ODE calculated as the mean absolute error (MAE) and shown on a logarithmic scale. (b) Plot showing the estimated value of the θ_m from which the geometric factor is determined as $\hat{\gamma}_1 = -\theta_m = 0.40$.

This methodological enhancement underscores the importance of ensemble approaches in neural network training and their contribution to robust model performance.

4.5.4. Robustness and generalization assessment

To evaluate the generalization capability of the Neural ODE model, testing was conducted using a broadband noise excitation that differs significantly from the training data. Random noise excitation contains energy distributed across a wide range of frequencies simultaneously and exhibits stochastic characteristics that fundamentally differ from the periodic structure of harmonic signals. This provides a rigorous evaluation of model generalization.

This study was performed numerically, as the experimental setup is limited to periodic excitation and cannot produce broadband random input signals. The Neural ODE was trained exclusively on harmonic excitation data (single-sine signals) synthetically generated by a numerical TLCD model according to Eq. (8). The natural frequency was assumed as $\omega_D = 0.64$ Hz based on preliminary studies with the specimen. The geometric factor γ_1, which corresponds to the modal mass, was kept at $\gamma_1 = 0.39$, while the head loss coefficient was set to $\delta = 8.275\ \mathrm{m}^{-1}$. As the test signal, zero-mean Gaussian white noise was applied, as shown in Fig. 13(a).

Despite this significant domain shift, the Neural ODE maintains satisfactory prediction accuracy for both displacement (Fig. 13(b)) and velocity (Fig. 13(c)) responses. While some discrepancies are visible — particularly in high-frequency components — the Neural ODE successfully captures the overall system dynamics, response amplitude, dominant frequency content, and phase relationships. These results

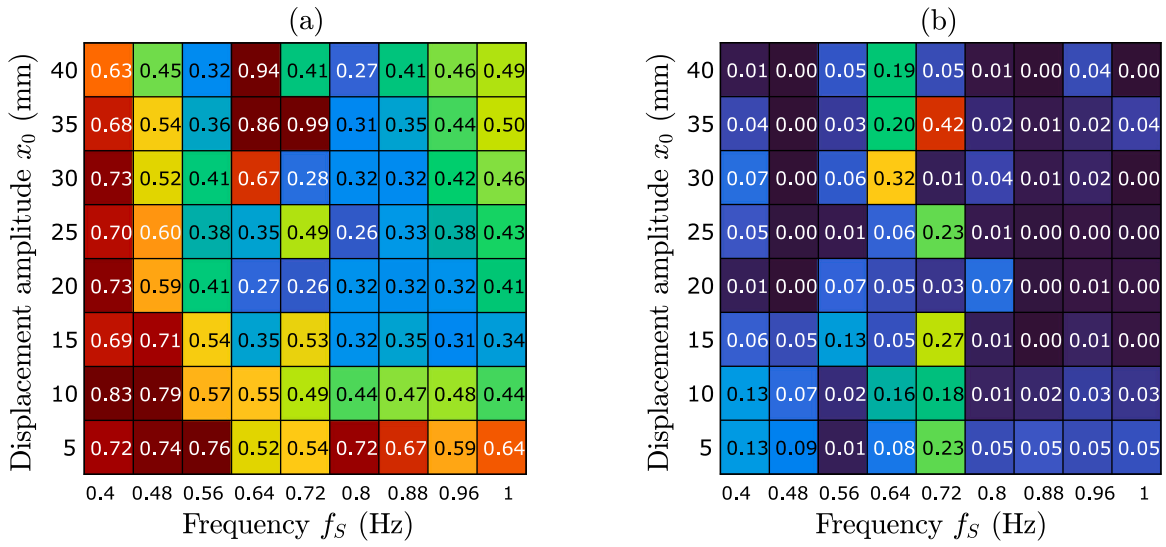

Fig. 12. The difference in the obtained mean absolute error (MAE) of the model when a single Neural ODE is used (a) and when an ensemble of Neural ODEs is used (b). The ensemble is used to reduce the variance error coming from the training algorithm.

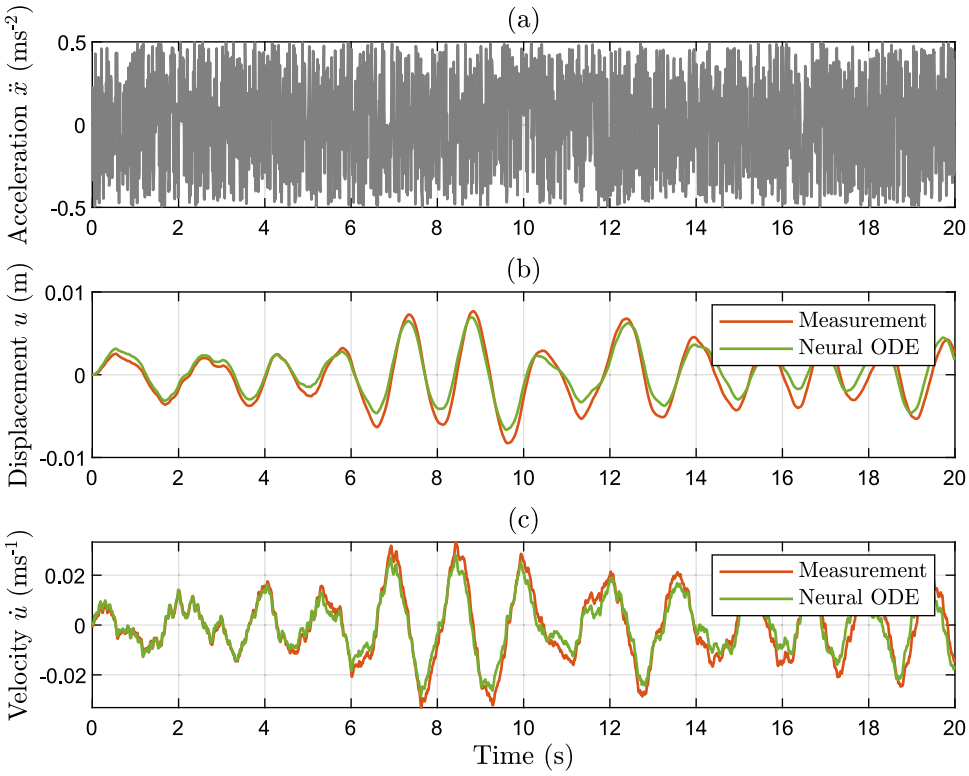

Fig. 13. Performance of the Neural ODE model in predicting broadband noise signal: (a) random noise excitation, (b) comparison of displacement responses, (c) comparison of velocity responses. The model demonstrates high accuracy despite the significant difference from the sine training signal.

demonstrate that the Neural ODE effectively learns the system dynamics and is capable of predicting responses to previously unseen excitation signals.

4.5.5. Sensitivity analysis of Neural ODE modifications

A sensitivity study was conducted to systematically isolate the effects of both the forced-input term and the modal mass parameter θ_m in the modified Neural ODE formulation. The study demonstrates the necessity of each component through controlled numerical experiments. Three distinct training scenarios were designed to isolate the individual contributions of forced excitation and modal mass parameterization, as shown in Table 4. All trained models were evaluated on identical sine-sweep reference data to ensure fair comparison of generalization performance.

The comparison between Cases 1 and 2 isolates the effect of the modal mass parameter when no forced excitation is present in training. As shown in Fig. 14 (Case 1), when the modal mass is provided, training on initial conditions alone produces excellent performance on the reference signal. The model successfully learns the system dynamics and accurately predicts both displacement and velocity responses. In contrast, Fig. 15 (Case 2) demonstrates the critical importance of the modal mass parameter. When θ_m is unknown, the model fails to achieve

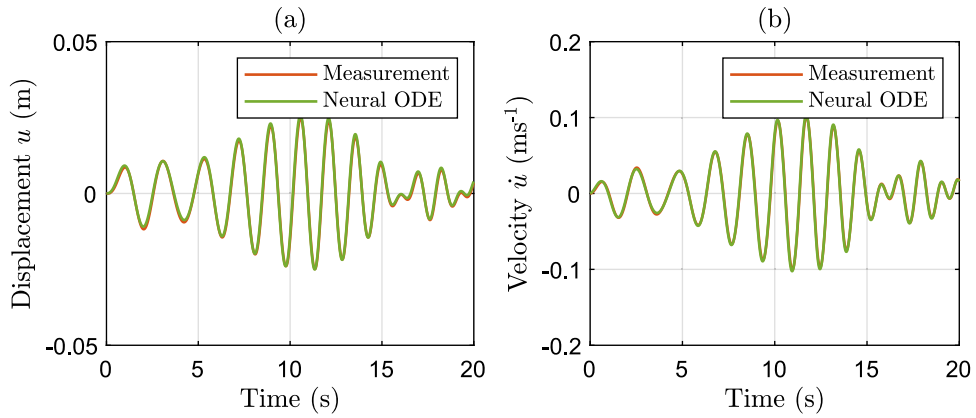

Fig. 14. Case 1 — training with initial conditions and known modal mass: comparison of displacement (a) and velocity (b) responses. Model is able to learn system dynamics from free response data.

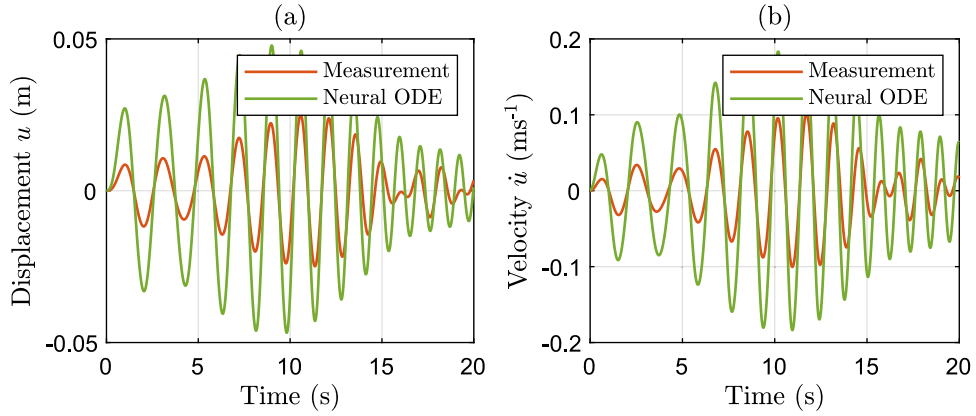

Fig. 15. Case 2 — training with initial conditions and unknown modal mass: comparison of displacement (a) and velocity (b) responses. Temporal patterns are captured, but amplitude scaling is incorrect due to missing mass parameter.

Table 3
Neural ODE configuration and training statistics for the optimal training dataset.

	Parameter	Value
Architecture	Hidden layer neurons	20
	Activation function	Sigmoid
	Ensemble size	10
Training parameters	Maximum iterations	10^4
	Batch size	200
	Time-length parameter (N_t)	40
	Learning rate	0.01
	Gradient decay rate	0.9
	Optimizer	Adam
	Initializer	Glorot
Performance metrics	Final training loss (MAE)	0.11
	Final validation loss (MAE)	0.15
	Reference set error (MAE)	0.23
	Identified geometric factor ($\hat{\gamma}_1$)	0.40
	Theoretical geometric factor (γ_1)	0.39

Table 4
Sensitivity study design to isolate the effects of forced excitation and modal mass parameter identification.

Case	Training excitation	Modal mass (θ_m) known
1	Initial conditions	Yes
2	Initial conditions	No
3	Harmonic excitation	No

proper scaling of the response, despite capturing the general dynamic behavior. The predicted responses exhibit correct temporal patterns but incorrect amplitudes, confirming that the modal mass parameter is essential for accurate forced response prediction. In Fig. 16 (Case 3), which corresponds to the proposed active learning framework, the model is able to both learn the system dynamics and discover the modal mass.

The training dynamics were further analyzed when θ_m is learned simultaneously with the neural network parameters. Fig. 17 compares the training loss convergence with and without modal mass parameter learning. The results show that including θ_m as a learnable parameter introduces only a marginal increase in training iterations, with no significant computational overhead, demonstrating the efficiency of the joint optimization approach. The convergence history of the modal mass parameter during training is shown in Fig. 18. The parameter consistently converges to the correct value across multiple training runs, confirming the stability and reliability of the identification process when using informative excitation signals.

The sensitivity study establishes several critical findings. First, the comparison of Cases 1 and 2 confirms that θ_m is essential for accurate amplitude prediction in forced systems. Second, the comparison of Cases 2 and 3 demonstrates that forced excitation data is necessary for modal mass identification. Third, the simultaneous learning of network parameters and θ_m is computationally efficient and numerically stable. Finally, the modal mass parameter consistently converges to the correct value when trained with informative excitation signals. These results validate both modifications to the standard Neural ODE formulation as necessary and sufficient for accurate dynamical system identification with unknown mass parameters.

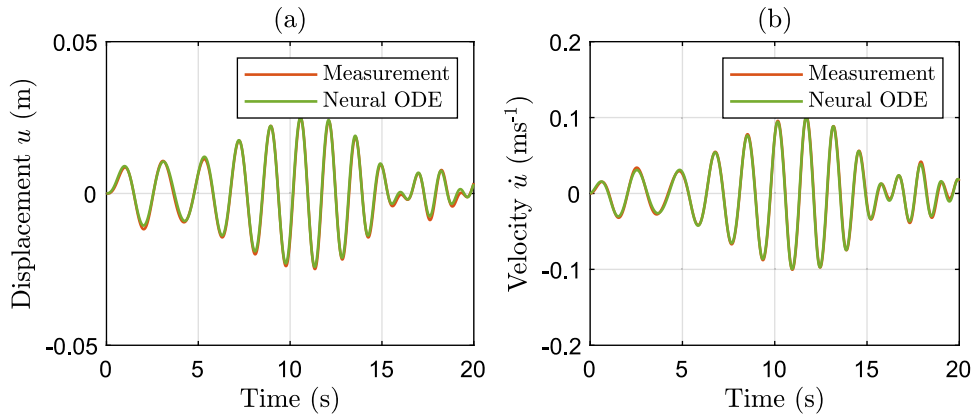

Fig. 16. Case 3 — training with harmonic excitation and unknown modal mass: comparison of displacement (a) and velocity (b) responses. Model is able to learn system dynamics and discover modal mass.

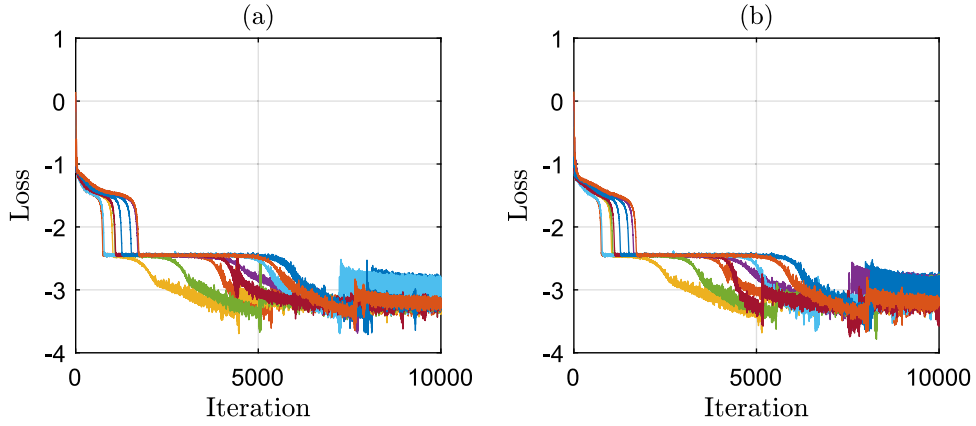

Fig. 17. Training loss convergence comparison with (a) and without (b) modal mass parameter learning. Each color represents a run with a different initial value. The identification of the modal mass introduces minimal additional computational cost.

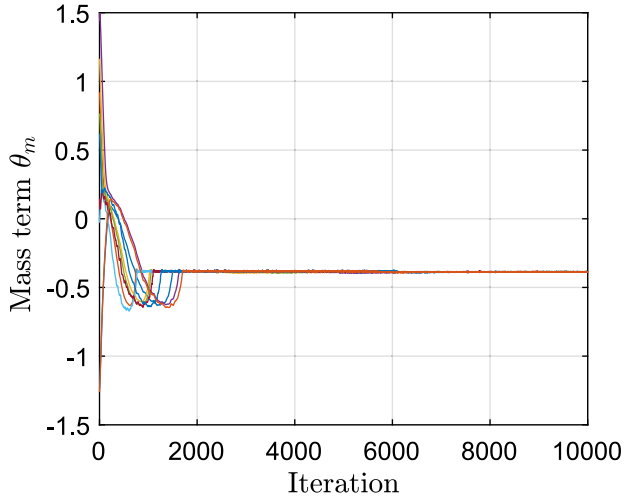

Fig. 18. Modal mass parameter convergence history during training across multiple runs with varying initial values (different colors). The consistent convergence to the correct value, regardless of initialization, demonstrates the reliability and robustness of the identification process.

4.5.6. Sensitivity analysis of optimization parameters

A sensitivity analysis of the pattern search optimization parameters was conducted to assess the robustness of the optimization procedure and to justify the selected algorithm settings. All experiments were conducted on the same TLCD identification problem, where the error surface shown in Fig. 7 was used as the objective function for the optimization. Similarly to results presented in Section 4.5, the success rate was used as the main performance metric. The analysis systematically examined three key parameters:

- *Initial mesh size:* 0.5, 1, 2, 4, and 8 (in scaled parameter space $[0, 10]$)
- *Poll method:* Four different polling strategies were tested, varying in the number and direction of search points evaluated at each iteration
- *Initial guesses:* 1000 initial points uniformly distributed using Halton Sequence Sampling

Fig. 19(a) shows the sensitivity to initial mesh size. The initial mesh size significantly affects optimization performance, with the optimal mesh size of 4 achieving the best overall performance, validating the choice used in the paper. Medium-range mesh sizes (2–4) provide optimal balance between exploration capability and convergence precision. In contrast, very small mesh sizes (0.5) lead to premature local convergence. This result is consistent with previous findings, confirming that intermediate mesh sizes enable effective parameter space coverage during optimization.

Fig. 19(b) shows a comparison of poll methods. The investigated four poll methods differ in how they generate candidate points around the current best solution. Poll Method #1 evaluates 3 points using a minimal positive basis with directions along the positive coordinate axes and one negative diagonal direction. Poll Method #2 evaluates

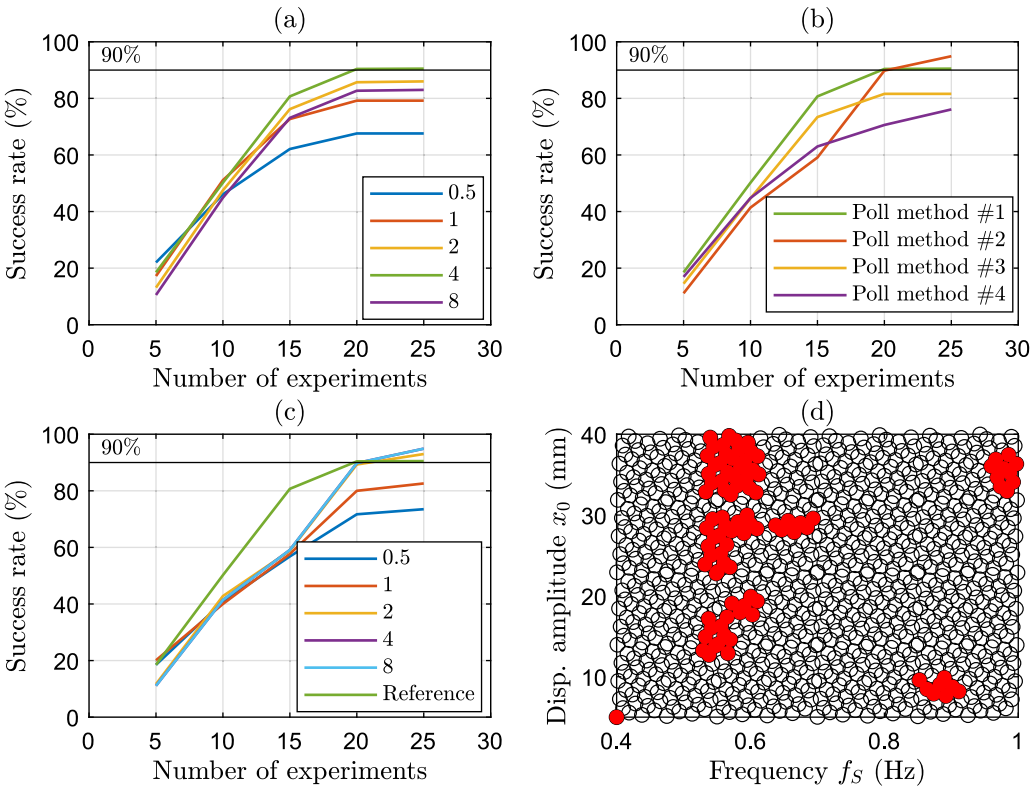

Fig. 19. Sensitivity analysis of pattern search optimization parameters: (a) influence of initial mesh size on convergence behavior; (b) comparison of different poll methods; (c) combined influence of poll method and mesh size; (d) spatial distribution of converged (hollow) and non-converged (red) initial points in the parameter space.

4 points using a positive basis that includes all positive and negative coordinate directions (i.e., $2N$ directions in an N-dimensional space). Poll Methods #3 and #4 employ adaptive mesh strategies that dynamically adjust the search directions based on local geometry: Poll Method #3 uses $N + 1$ directions with Mesh Adaptive Direct Search, while Poll Method #4 uses an orthogonalized mesh with $2N$ directions, enabling a more systematic exploration of the search space. Results show that the choice of poll method can significantly affect the performance of the algorithm. Poll Method #2 achieves the highest overall success rate but requires more experiments. Poll Method #1, used in Section 4.5, reaches 90% success rate with fewer average function evaluations, providing better efficiency. Poll Method #1 uses 3 poll points versus 4 for Poll Method #2, explaining the efficiency advantage when fewer experiments are required. Both Poll Methods #3 and #4 show inferior performance for this specific problem. Hence, the choice of Poll Method #1 represents an optimal balance between reliability and computational efficiency.

In Fig. 19(c), Poll Method #2 was further evaluated with different initial mesh sizes to examine whether its performance could be improved. Testing Poll Method #2 with varying initial mesh sizes confirms that the influence of mesh size effect is consistent across poll methods. Nevertheless, the reference configuration (Poll Method #1, mesh size of 4) yields the best overall performance.

Finally, Fig. 19(d) presents the initial guess distribution analysis. The spatial distribution of converged and non-converged initial points reveals a clear correlation between proximity to the global optimum and convergence success. Initial points in the range of $[0.5 \dots 0.6]$ Hz exhibit higher failure rates, likely due to mesh size causing the algorithm to overshoot potential solutions in this region. A few scattered non-converged points are observed, which can be attributed to the stochastic nature of the optimization landscape. It should be noted that these initial points were also used to obtain the convergence statistics shown in Fig. 10.

The findings confirm that the chosen pattern search configuration was deliberately selected to minimize function evaluations while maintaining effectiveness. The sensitivity analysis validates this choice by showing that the selected parameters (mesh size 4, Poll Method #1) provide near-optimal performance. Further tuning of these parameters yields marginal improvements in success rate, but at the cost of a significantly higher number of function evaluations, i.e., experiments. As noted, the parameter space was scaled to $[0, 10]$ for both amplitude and frequency to ensure algorithmic generality across different TLCD configurations. This scaling approach, combined with the validated parameter settings, establishes a robust optimization framework that can be translated to other problems.

4.5.7. Comparison with a genetic algorithm

For the proposed active learning framework, pattern search was selected due to its demonstrated efficiency in other applications with respect to the number of required function evaluations, which in this context correspond directly to physical experiments. While more advanced optimization algorithms could also be used when experimental budgets and timelines allow, an algorithm such as pattern search offers clear advantages when each function evaluation requires substantial time and resources. To justify the choice of optimization algorithm, a comparison between pattern search and a genetic algorithm was conducted using identical problem setups and convergence criteria, as described in Section 4.5.2. Both algorithms were tested in 1000 independent optimization runs to ensure statistical significance.

The results in Fig. 20 demonstrate clear advantages of pattern search for this application. Both algorithms were configured to achieve a success rate of 90% across 1000 runs to ensure a fair comparison. Fig. 20(a) shows the distribution of function evaluations for both algorithms. The pattern search exhibits a more concentrated distribution around the median value, whereas the genetic algorithm shows

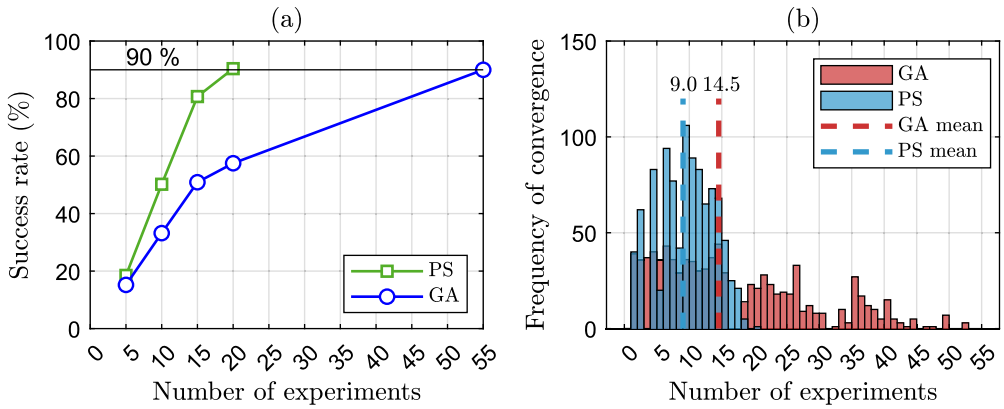

Fig. 20. (a) Comparison of the success rates of the pattern search (PS) and the genetic algorithm (GA) (b) histogram showing the frequency of convergence for the pattern search and the genetic algorithm for 1000 runs.

Table 5
Comparison of pattern search and genetic algorithm performance over 1000 independent optimization runs. Pattern search demonstrates superior efficiency and lower variability.

Metric	Pattern search	Genetic algorithm
Mean function evaluations	9.0	14.5
Median function evaluations	9.0	13.0
Standard deviation	4.2	11.9
Success rate at 90%	20 experiments	55 experiments

substantially greater variability, as reflected by its higher standard deviation and a larger number of outliers that require excessive function evaluations. This variability makes pattern search more predictable and therefore better suited for experimental planning, where resource allocation and time scheduling are critical constraints. Table 5 presents a quantitative comparison of the two optimization algorithms. Quantitatively, pattern search requires significantly fewer function evaluations on average, with a mean of 9.0 and a standard deviation of 4.2, compared to the genetic algorithm's mean of 14.5 and a standard deviation of 11.9. The median number of function evaluations further highlights this efficiency advantage: Pattern search required only 9.0 evaluations compared to 13.0 for the genetic algorithm, representing approximately a 31% reduction in experimental cost. This empirical comparison validates the choice of pattern search as the optimization algorithm for the proposed framework, demonstrating its superior efficiency and reliability for experimental parameter identification in this application.

5. Application example

In this section, we present how the obtained Neural ODE model of the TLCD can be further utilized. Here, the focus is on its application in structural analysis, but further applications such as structural health monitoring are possible as well. Our goal is to assess whether the damper properties are appropriate for the structure. In particular, we investigate an SDOF system subjected to a dynamic force shown in Fig. 21. The application process is similar for multi-degree-of-freedom (MDOF) systems with either single or multiple TLCDs.

The equation of motion representing the structure reads as follows:

$$\ddot{x} + 2D_s\omega_s\dot{x} + \omega_s^2 x = -\mu(\ddot{x} + \hat{\gamma}_2\hat{\ddot{u}}) + f, \tag{10}$$

where the damping ratio of the structure is $D_s = 1\%$, the natural frequency is $\omega_s = 4.22$ rad s^{-1} corresponding to 0.67 Hz, and the mass ratio of the damper liquid mass to the modal mass of the structure is $\mu = m_d/m_s = 11.7/180 = 6.5\%$. Additionally, $f = F(t)/m_s$ represents the time-dependent, mass-normalized excitation. Corresponding to the

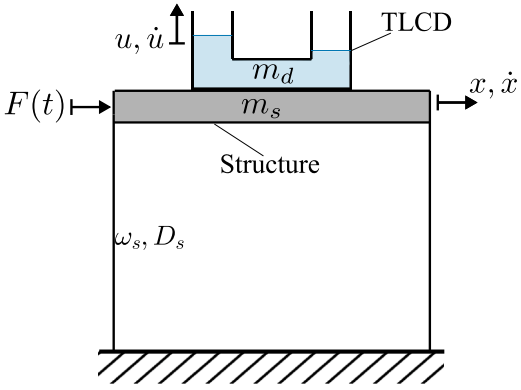

Fig. 21. The studied structure with the TLCD.

reference signal, with which the Neural ODE model was obtained, a sine-sweep signal is chosen in the range of 0.40–1.00 Hz as excitation with constant amplitude, as shown in Fig. 22(a). In Fig. 22(b), it can be seen that the frequency of the excitation signal increases linearly over time.

When coupled with the structure, a second geometric factor, $\hat{\gamma}_2$, has to be estimated. Similar to the first geometric factor, γ_1, it depends on the TLCD's geometry. Since the cross-sectional areas, A_V and A_H, can be easily measured, it is assumed that they are known. However, as mentioned before, L_H and L_V are approximations of the liquid flow direction and are particularly challenging to determine for complex TLCD layouts. Therefore, we assume L_H based on the TLCD's geometry and estimate the missing L_V substituting the first geometric factor, $\hat{\gamma}_1$, estimated by the Neural ODE in the corresponding Eq. (9). Once L_V is identified, the second geometric factor can be estimated from:

$$\hat{\gamma}_2 = \frac{L_H}{2L_V + \frac{A_H}{A_V}L_H}. \tag{11}$$

In this way, any error introduced by the approximation of L_H is effectively removed. In Eq. (10), the acceleration response of the TLCD, $\hat{\ddot{u}}$, is estimated by the Neural ODE model. Corresponding to Eq. (8), this model uses the acceleration of the structure, $\ddot{x}$, multiplied by the first geometric factor, $\hat{\gamma}_1$, as its dynamic force input. A simultaneous inference of the Neural ODE model with Eq. (10) yields the dynamic response of the structure-TLCD system. It can be seen from Fig. 22(d) that the frequency response of the structure is significantly improved when compared to the uncontrolled case. This is further highlighted in Fig. 22(c), which shows the displacement time histories of the structure with and without the TLCD.

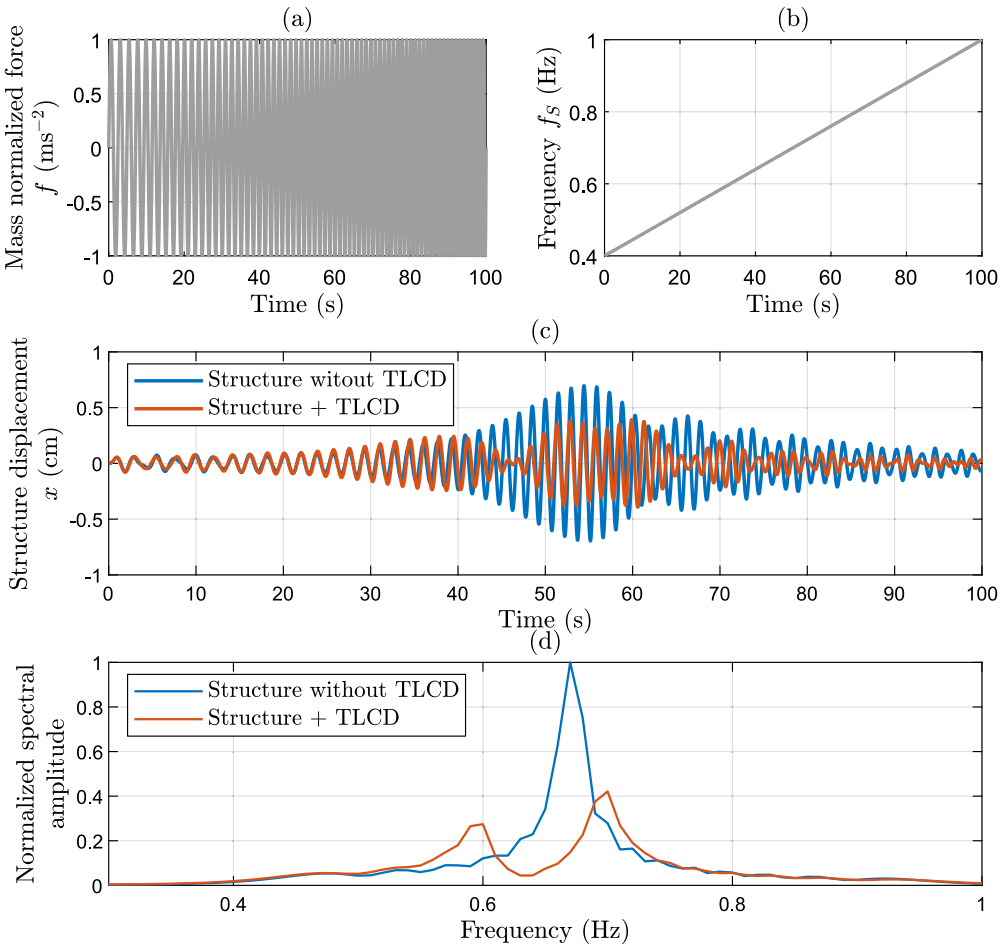

Fig. 22. (a) Excitation signal. (b) The frequency content of the excitation signal plotted over time. (c) Displacement of the structure. (d) The frequency response of the structure with and without the damper.

6. Conclusion

In this study, an active learning framework was proposed for data-efficient modeling of TMDs. The framework aims to efficiently generate experimental data and develop a model of the tested specimen. It allows the model to actively query the parameters defining the experimental conditions, such as the excitation signal, ambient temperature, and factors specific to the specimen. For this purpose, the pattern search algorithm was employed to optimize the experimental parameters, focusing on improving model accuracy by generating the most informative data with the fewest experiments. To evaluate the model, a reference signal was used, corresponding to those typically applied in laboratory performance testing based on application requirements and industry codes. As a model, extended versions of Neural ODEs were developed, enabling the prediction of system states under forced vibrations without requiring restoring force measurements. Unlike their classical counterparts, the proposed Neural ODE architecture can also estimate the modal mass of the control device in addition to its state. This new feature is expected to be useful not only for the paper's target application but also for other structural dynamics modeling and system identification tasks. By employing an ensemble of Neural ODEs, the training variance was reduced, further enhancing the framework's performance.

The proposed framework was validated on a TLCD prototype mounted on a shaking table, as it would be for an industrial performance test. A simple harmonic signal was used to generate response data, with the amplitude and frequency selected as experimental parameters. A sine sweep was used as the reference signal. After randomly initializing the optimization algorithm, the proposed framework identified the optimal training dataset. Once trained on this dataset, the Neural ODE model accurately predicted the TLCD response, demonstrating the framework's effectiveness. The framework was then compared to three unsupervised sampling methods (Uniform Random Sampling, LHS, Sobol Sequence Sampling). The proposed framework outperformed these methods, achieving a success rate of 90% with only 20 experiments, whereas the best-performing unsupervised sampling method (Sobol sequence) required at least 35 experiments to reach the same success rate, representing a 43% reduction in required experimental effort. Finally, the generated Neural ODE model was used to simulate the TLCD when coupled with a structure, showing the potential of the generated model in engineering applications.

6.1. Contributions of the proposed framework to research and industrial communities

6.1.1. Scientific contributions

- Development of an active learning framework that autonomously selects informative experiments via pattern search to maximize model accuracy with minimal experiments.
- Extension of the Neural ODE architecture to forced vibrations which includes modal mass estimation without restoring-force measurements.
- Experimental validation demonstrating that the proposed method achieves the target accuracy with significantly fewer experiments than the best unsupervised sampling methods.

6.1.2. Industrial relevance

- Accelerated design and control analysis for TMDs and TLCDs through reduced experimental effort.
- Reduced instrumentation requirements by eliminating the need for force measurements.
- Seamless integration with standard performance testing protocols.
- Efficient development of digital twins for vibration control devices. The resulting data-driven models can be readily applied to control system design, performance optimization, and predictive maintenance applications.

6.2. Limitations of the study

The current framework is designed for SDOF systems, accommodating only nonlinearities with weak memory effects, where the response characteristics do not strongly depend on past states. While this is sufficient for the target application, future research could focus on expanding the framework to other systems. Such an expansion will necessitate the development of additional training strategies, including the use of additive training datasets. Furthermore, the approach does not provide direct uncertainty quantification of the trained model. Neural ODEs do not inherently quantify prediction uncertainty, which limits confidence assessment of model predictions. While ensemble methods can partially address this limitation by providing variability estimates across models, formal uncertainty quantification remains an area for future development.

Glossary

List of Abbreviations

Adam Adaptive Moment Estimation

ANN Artificial neural network

DOE Design of experiments

FNN Feedforward neural network

GPSSMs Gaussian Process State-Space Models

LHS Latin Hypercube Sampling

MAE Mean absolute error

MDOF Multi degree of freedom

Neural ODE Neural Ordinary Differential Equation

ODE Ordinary differential equation

SDOF Single degree of freedom

TLCD Tuned liquid column damper

TMD Tuned mass damper

CRediT authorship contribution statement

Pavle Milicevic: Writing – original draft, Visualization, Validation, Investigation, Formal analysis. **Sebastian Stemmler:** Writing – review & editing, Resources, Funding acquisition. **Okyay Altay:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to express their sincere appreciation for the financial support from the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) with the grant number: 538131242. The authors would also like to thank Dr.-Ing. Markus Zimmer, for his support during the preparation of the experimental setup.

Data availability

Data will be made available on request.

References

Abbiati, G., Marelli, S., Ligeikis, C., Christenson, R., Stojadinović, B., 2022. Training of a classifier for structural component failure based on hybrid simulation and kriging. J. Eng. Mech. 148 (1), http://dx.doi.org/10.1061/(ASCE)EM.1943-7889.0002048.

Altay, O., 2021. Vibration Mitigation Systems in Structural Engineering. CRC Press, Boca Raton, Florida, http://dx.doi.org/10.1201/9781315122243.

Altay, O., Klinkel, S., 2018. A semi-active tuned liquid column damper for lateral vibration control of high-rise structures: Theory and experimental verification. Struct. Control. Health Monit. 25 (12), e2270. http://dx.doi.org/10.1002/stc.2270.

Bradley, W., Volkovinsky, R., Boukouvala, F., 2024. Enabling global interpolation, derivative estimation and model identification from sparse multi-experiment time series data via neural ODEs. Eng. Appl. Artif. Intell. 130, 107611. http://dx.doi.org/10.1016/j.engappai.2023.107611.

Bull, L., Rogers, T., Wickramarachchi, C., Cross, E., Worden, K., Dervilis, N., 2019. Probabilistic active learning: An online framework for structural health monitoring. Mech. Syst. Signal Process. 134, 106294. http://dx.doi.org/10.1016/j.ymssp.2019.106294.

Chen, T.Q., Rubanova, Y., Bettencourt, J., Duvenaud, D.K., 2018. Neural ordinary differential equations. In: 32nd Conference on Neural Information Processing Systems. Montréal, Canada, pp. 6571–6583.

Chen, Y., Zhao, L., Han, T., 2025. Uncertainty-guided Bayesian active learning for cost-effective fault diagnosis with minimal labeled data. Eng. Appl. Artif. Intell. 160, 111945. http://dx.doi.org/10.1016/j.engappai.2025.111945.

Cunha, B.Z., Droz, C., Zine, A.-M., Foulard, S., Ichchou, M., 2023. A review of machine learning methods applied to structural dynamics and vibroacoustic. Mech. Syst. Signal Process. 200, 11035. http://dx.doi.org/10.1016/j.ymssp.2023.110535.

Fisher, R.A., 1990. Statistical Methods, Experimental Design and Scientific Inference: A Re-Issue of Statistical Methods for Research Workers, The Design of Experiments and Statistical Methods and Statistical Inference. Oxford University Press, Oxford.

Furtmüller, T., Di Matteo, A., Adam, C., Pirrotta, A., 2019. Base-isolated structure equipped with tuned liquid column damper: An experimental study. Mech. Syst. Signal Process. 116, 816–831. http://dx.doi.org/10.1016/j.ymssp.2018.06.048.

Ganaie, M., Hu, M., Malik, A., Tanveer, M., Suganthan, P., 2022. Ensemble deep learning: A review. Eng. Appl. Artif. Intell. 115, 105151. http://dx.doi.org/10.1016/j.engappai.2022.105151.

Gardner, P., Dal Borgo, M., Ruffini, V., Hughes, A., Zhu, Y.-C., Wagg, D., 2020. Towards the development of an operational digital twin. Vibration 3 (3), 235–265. http://dx.doi.org/10.3390/vibration3030018.

Guo, Q., Yang, X., Li, K., Li, D., 2025. Parameters identification of magnetorheological damper based on particle swarm optimization algorithm. Eng. Appl. Artif. Intell. 143, 110016. http://dx.doi.org/10.1016/j.engappai.2025.110016.

Hametner, C., Stadlbauer, M., Deregnaucourt, M., Jakubek, S., Winsel, T., 2013. Optimal experiment design based on local model networks and multilayer perceptron networks. Eng. Appl. Artif. Intell. 26 (1), 251–261. http://dx.doi.org/10.1016/j.engappai.2012.05.016.

Hughes, A., Bull, L., Gardner, P., Dervilis, N., Worden, K., 2022. On robust risk-based active-learning algorithms for enhanced decision support. Mech. Syst. Signal Process. 181, 109502. http://dx.doi.org/10.1016/j.ymssp.2022.109502.

Kiani, J., Camp, C., Pezeshk, S., Khoshnevis, N., 2020. Application of pool-based active learning in reducing the number of required response history analyses. Comput. Struct. 241, 106355. http://dx.doi.org/10.1016/j.compstruc.2020.106355.

Lai, Z., Mylonas, C., Nagarajaiah, S., Chatzi, E., 2021. Structural identification with physics-informed neural ordinary differential equations. J. Sound Vib. 508, 116196. http://dx.doi.org/10.1016/j.jsv.2021.116196.

Mehrkian, B., Altay, O., 2020. Mathematical modeling and optimization scheme for omnidirectional tuned liquid column dampers. J. Sound Vib. 484, 115523. http://dx.doi.org/10.1016/j.jsv.2020.115523.

Mehrkian, B., Altay, O., 2025. Semi-active omnidirectional liquid column vibration absorber with rapid frequency adjustment capability. Mech. Syst. Signal Process. 224, 112093. http://dx.doi.org/10.1016/j.ymssp.2024.112093.

Milicevic, P., Altay, O., 2024. Data generation framework for inverse modeling of nonlinear systems in structural dynamics applications. Acta Mech. 235 (3), 1493–1515. http://dx.doi.org/10.1007/s00707-023-03532-3.

Nelles, O., 2020. Nonlinear Dynamic System Identification. Springer, Cham, http://dx.doi.org/10.1007/978-3-030-47439-3.

Pei, J.-S., Hougen, D.F., Kanneganti, S.T., Wright, J.P., Mai, E.C., Smyth, A.W., Masri, S.F., Derkevorkian, A., Gay-Balmaz, F., Komini, L., 2022. Interpretable machine learning for function approximation in structural health monitoring. In: Cury, A., Ribeiro, D., Ubertini, F., Todd, M.D. (Eds.), Structural Health Monitoring Based on Data Science Techniques. Springer, pp. 369–388. http://dx.doi.org/10.1007/978-3-030-81716-9_18.

Rapp, J.T., Bremer, B.J., Romero, P.A., 2024. Self-driving laboratories to autonomously navigate the protein fitness landscape. Nat. Chem. Eng. 1 (1), 97–107. http://dx.doi.org/10.1038/s44286-023-00002-4.

Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Gupta, B.B., Chen, X., Wang, X., 2021. A survey of deep active learning. ACM Comput. Surv. 54 (9), http://dx.doi.org/10.1145/3472291.

Samoa, P., Aronsson, L., Longa, A., Leitner, P., Chehreghani, M.H., 2024. A unified active learning framework for annotating graph data for regression tasks. Eng. Appl. Artif. Intell. 138, 109383. http://dx.doi.org/10.1016/j.engappai.2024.109383.

Settles, B., 2009. Active learning literature survey. In: Computer Sciences, Technical Report 1648, University of Wisconsin, Madison, Wisconsin.

Settles, B., 2012. Active learning. In: Synthesis Lectures on Artificial Intelligence and Machine Learning, Springer, Cham, http://dx.doi.org/10.1007/978-3-031-01560-1.

Shields, M.D., Gurley, K., Catarelli, R., Chauhan, M., Ojeda-Tuz, M., Masters, F.J., 2023. Active learning applied to automated physical systems increases the rate of discovery. Sci. Rep. 13 (1), 8402. http://dx.doi.org/10.1038/s41598-023-35257-7.

Sobot, T., Stankovic, V., Stankovic, L., 2024. Human in the loop active learning for time-series electrical measurement data. Eng. Appl. Artif. Intell. 133, 108589. http://dx.doi.org/10.1016/j.engappai.2024.108589.

Torczon, V., 1997. On the convegence of pattern search algorithms. SIAM J. Optim. 7 (1), 1–25. http://dx.doi.org/10.1137/S1052623493250780.

Yang, Y., Loog, M., 2018. A benchmark and comparison of active learning for logistic regression. Pattern Recognit. 83, 401–415. http://dx.doi.org/10.1016/j.patcog.2018.06.004.

Yu, H.S.A., Yao, D., Zimmer, C., Toussaint, M., Nguyen-Tuong, D., 2021. Active learning in Gaussian process state space model. In: Oliver, N., Pérez-Cruz, F., Kramer, S., Read, J., Lozano, J.A. (Eds.), Machine Learning and Knowledge Discovery in Databases. Research Track. ECML PKDD 2021, In: Lecture Notes in Computer Science, Springer International Publishing, Cham, pp. 346–361. http://dx.doi.org/10.1007/978-3-030-86523-8_21.

Yuan, Y., Au, F.T., Yang, D., Zhang, J., 2023. Active learning structural model updating of a multisensory system based on kriging method and Bayesian inference. Comput.-Aided Civ. Infrastruct. Eng. 38 (3), 353–371. http://dx.doi.org/10.1111/mice.12822.

Zhang, W.-J., Yuen, K.-V., Yan, W.-J., 2023. Active learning aided Bayesian nonparametric general regression for model updating using modal data. Mech. Syst. Signal Process. 204, 110830. http://dx.doi.org/10.1016/j.ymssp.2023.110830.