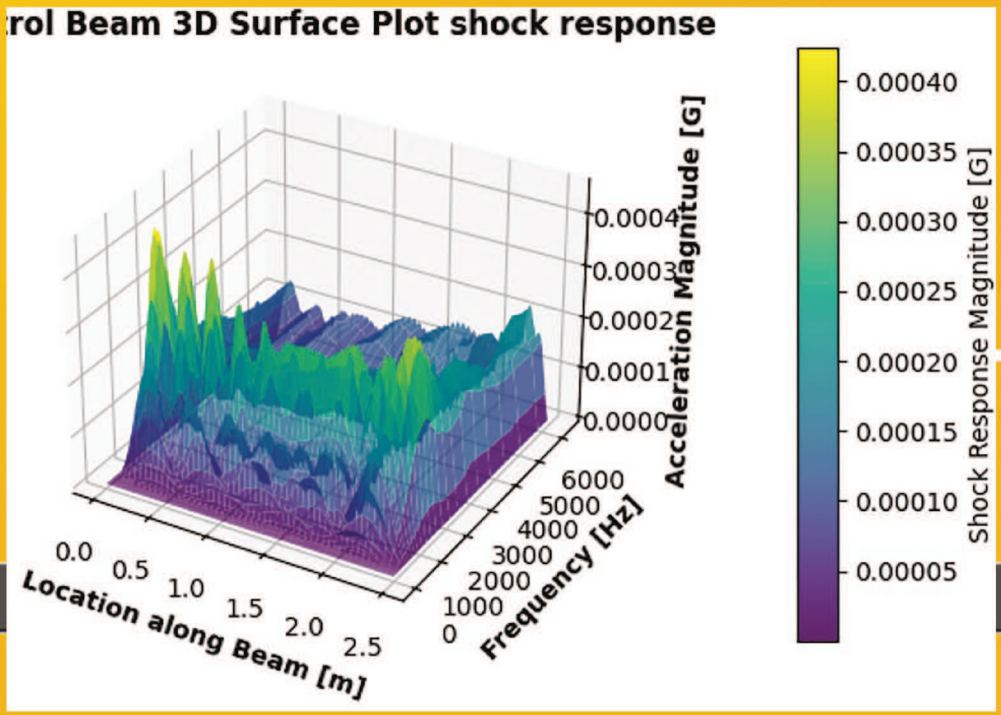
River Rapids

Conference Proceedings of the Society for Experimental Mechanics Series

Data Science in Engineering, Volume 9

Thomas Matarazzo
François Hemez
Eleonora Maria Tronci
Austin Downey



Proceedings of the 44th IMAC, A Conference and Exposition on Structural Dynamics 2026

River Publishers

Conference Proceedings of the Society for Experimental Mechanics Series

Series Editor

Kristin B. Zimmerman
Society for Experimental Mechanics, Inc.,
Bethel, USA

The Conference Proceedings of the Society for Experimental Mechanics Series presents early findings and case studies from a wide range of fundamental and applied work across the broad range of fields that comprise Experimental Mechanics. Series volumes follow the principle tracks or focus topics featured in each of the Society's two annual conferences: IMAC, A Conference and Exposition on Structural Dynamics, and the Society's Annual Conference & Exposition and will address critical areas of interest to researchers and design engineers working in all areas of Structural Dynamics, Solid Mechanics and Materials Research.

Thomas Matarazzo · François Hemez · Eleonora Maria Tronci · Austin Downey
Editors

Data Science in Engineering, 2026, Vol. 9

Proceedings of the 44th IMAC, A Conference and Exposition on Structural Dynamics 2026



Published, sold and distributed by:

River Publishers
Broagervej 10
9260 Gistrup
Denmark

www.riverpublishers.com

ISBN 97887-438-1448-1 (Hardback)

ISBN 97887-438-1420-7 (eBook)

https://doi.org/10.13052/rp-9788743814207

Conference Proceedings of the Society for Experimental Mechanics

An imprint of River Publishers

© The Society for Experimental Mechanics, Inc. 2026

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, or reproduction in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use. The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Preface

Data Science in Engineering represents one of ten volumes of technical papers presented at the 44th IMAC, A Conference and Exposition on Structural Dynamics, organized by the Society for Experimental Mechanics and held January 19-22, 2026. The full proceedings also include volumes on Nonlinear Structures & Systems; Dynamics of Civil Structures; Model Validation and Uncertainty Quantification; Dynamic Substructuring & Transfer Path Analysis; Special Topics & Experimental Techniques in Structural Dynamics; Computer Vision & Laser Vibrometry for Structural Dynamics; Dynamic Environments and Durability Testing; Topics in Modal Analysis & Parameter Identification in Structural Dynamics; and Structural Health Monitoring, Metamaterials, Metastructures & Additive Manufacturing and Dynamics of Biomedical and Bioinspired Systems.

The Data Science in Engineering technical division of the Society for Experimental Mechanics focuses on the transformative role and application of data analytics and modeling in structural, aerospace, and mechanical engineering. With the rapid advancements in sensing technologies (high-speed cameras, image processing, laser sensors, unmanned aerial vehicles, etc.) and high-performance computing, engineering systems now generate vast volumes of data that demand advanced analytical frameworks to manage big data and extract useful information. Machine learning, deep learning, transfer learning, big data, statistics, and related methods define the analytical toolset to process vast volumes of measurements and predictions, analyze complex phenomena, identify trends and relationships, and guide the development of predictive models through data. Statistical and machine learning methods provide fast, robust, adaptive, and scalable solutions for the online monitoring of structures and mechanical systems, and to support decision-making and risk analysis. These advancements are transforming how engineers collect and process data, shifting from reactive assessments to proactive strategies that enhance system understanding, performance, resilience, and operational efficiency.

The organizers would like to thank the authors, presenters, session organizers, and session chairs for their participation in this track.

Editors:

Thomas Matarazzo – United States Military Academy;
François Hemez – Lawrence Livermore National Laboratory, Livermore, CA, USA;
Eleonora Maria Tronci – New York University;
Austin Downey – University of South Carolina

Contents

1 Full-Field 3D Structural Dynamics via Neural Radiance Fields 1
Thiago F. Ribeiro, Victor H. R. Cardoso, João C. W. A. Costa, and Moisés F. Silva

2 Investigation on the identification of Spatially-Varying Elastic and Damping Properties in Beams and Plates Using Physics-Informed Neural Networks 17
Morvan Ouisse, Maxime Auger, João Gabriel Oliveira Aveiro, Roberta Tittarelli, Valentin Calisti, Rafael Teloli, Emmanuel Ramasso, Patrice Le Moal

3 Neural Operators for Accelerating Flow Field Predictions in an Empty Duct 25
Alan Xavier and Ludovic Renson

4 Shaping a Shock Response Spectrum via Metamaterial Topology Optimization 31
Brannon Boyd and Vijaya V. N. Sriram Malladi

5 Physics-based Online Cyber-Physical Neural Network Models for Real-time Hybrid Simulation 41
Faisal Malik, Liang Cao, James Ricles, and Austin Downey

6 Physics-Informed Machine Learning Part III: Hard-Constraint ODE Method for Structural Dynamics 49
Mohsen Gol Zardian, Austin R.J. Downey, Eleonora Maria Tronci, Conor Madden, Daniel Coble, Sina Navidi, and Chao Hu

7 Multi-Task Equation Discovery 61
S.C. Bee, K. Worden, N. Dervilis, and L.A. Bull

8 Physics-Informed Machine Learning Part IV: Weight-Tuned Soft-Constraint Method for Structural Dynamics 69
Eleonora Maria Tronci, Austin R. J. Downey, Connor Madden, Mohsen Gol Zardian, and Daniel Coble

9 Temperature-driven Unsupervised Damage Identification using Cepstral Features 81
Lorenzo Sclafani, Lorenzo Stagi, Silvia Milana, and Eleonora M. Tronci

10 Leveraging Agentic AI and LLMs to Enhance Conceptual Understanding of Structural Dynamics 93
Krishna Teja Josyula and Vijaya V. N. Sriram Malladi

11 AI-Enhanced Anomaly Detection of Acoustic Emission Sensor Data via Clustering and Noise Discrimination: A Machine Learning-Based Study 103
Ali Bakhtiari, Tonghao Zhang, Antonio De Luca, and Nezar Abraham

12 On the Communities of a Population-based SHM Network 113
Daniel S. Brennan and Keith Worden

Chapter 1
Full-Field 3D Structural Dynamics via Neural Radiance Fields

Thiago F. Ribeiro, Victor H. R. Cardoso, João C. W. A. Costa, and Moisés F. Silva

Abstract The ability to acquire high-resolution, full-field 3D representations of structures has long enabled experimental modal identification, whether via contact sensors that track material points in a Lagrangian frame or optical imagers that sample motion in an approximate Eulerian frame. Conventional algorithms assume fixed measurement identities through time, but large-motion dynamics, occlusions, viewpoint changes, and noise violate both frames and undermine standard techniques. We introduce the first Neural Radiance Field (NeRF)-based pipeline for full-field 3D experimental modal analysis driven solely by standard RGB cameras. A semi-spherical array of synchronized cameras records the structure's oscillation at discrete time steps. At each step, we train a NeRF to reconstruct the instantaneous 3D light field and extract a dense point cloud with a Lagrangian interpretation. The resulting sequence of dynamically varying point clouds is then processed via principal component analysis for dimensionality reduction, followed by a blind source separation scheme to decompose modal frequencies and recover 3D mode shapes. Validation on simulated data yields modal parameters within experimental error margins of reference values, demonstrating accuracy on par with specialized hardware while relying solely on ubiquitous RGB imaging and neural reconstruction. By integrating NeRF reconstructions with classical modal-analysis workflows, this work establishes a cost-effective, vision-based alternative for structural diagnostics. Motivated by the very first attempt to assess the health condition of plants through their vibrational modes, treating them within the framework of structural health monitoring, this work also represents the first application of dynamic Neural Radiance Fields for full-field modal analysis of plants, enabling non-invasive extraction of quantitative physiological traits.

Keywords Neural radiance fields · 3D scanning · BSS · PCA · dynamic cloud points

Introduction

Modal analysis of dynamic structures represents a fundamental methodology in structural engineering and mechanical systems characterization, enabling the identification of natural frequencies, damping ratios, and mode shapes essential for design validation and health monitoring. Traditional contact-based sensor networks impose significant limitations in terms of spatial resolution, installation complexity, and applicability to inaccessible environments. The emergence of vision-based measurement techniques has revolutionized structural dynamics assessment by offering non-contact, high-resolution spatial measurements capable of capturing full-field deformation patterns [1, 2].

Recent advances in computer vision and neural implicit representations have introduced NeRFs as a powerful tool for 3D scene reconstruction and novel view synthesis. NeRFs achieve state-of-the-art results by representing scenes through fully-connected neural networks that map 5D coordinates (spatial location and viewing direction) to volume density and view-dependent radiance [3]. This implicit representation enables high-fidelity 3D reconstruction from sparse 2D observations, making it particularly suitable for capturing dynamic structural behavior with unprecedented spatial detail.

Thiago F. Ribeiro · Victor H. R. Cardoso · João C. W. A. Costa
Applied Electromagnetism Laboratory
Federal University of Pará, Belém, PA, Brazil
e-mail: thiago.figueiro.ribeiro@gmail.com; victorcard@ufpa.br; jweyl@ufpa.br

Moisés F. Silva
Los Alamos National Laboratory
Los Alamos, New Mexico 87544, USA
e-mail: cien.felipe@gmail.com

© The Author(s), under exclusive license to River Publishers 2026

Kristin B. Zimmerman, PhD (eds.), *Data Science in Engineering, 2026, Vol. 9*, Conference Proceedings
of the Society for Experimental Mechanics Series, https://doi.org/10.13052/rp-9788743814207

The application of 3D point cloud data to structural dynamics has gained significant momentum with the development of time-of-flight (ToF) imaging technologies. The work present in [4] demonstrated the feasibility of extracting vibration modes from dynamic point clouds using commercial depth sensors, employing virtual Lagrangian sensors to track structural motion and blind source separation techniques for modal parameter extraction. This pioneering work established that point cloud-based approaches could achieve high-resolution modal identification comparable to traditional laser displacement sensors, with the advantage of capturing thousands of measurement points simultaneously.

While traditional point cloud acquisition relies on sequential scanning or ToF sensors with limited temporal resolution, NeRF-based reconstruction offers distinct advantages for dynamic structural analysis. Unlike conventional photogrammetry or stereo vision approaches, NeRF technology provides superior handling of complex lighting conditions, view-dependent effects, and scene completeness. The continuous volumetric representation inherent in NeRFs enables the generation of dense, temporally consistent point clouds from multi-view imagery, addressing the fundamental challenge of maintaining spatial correspondence across time-varying datasets.

Principal Component Analysis (PCA) serves as the mathematical foundation for extracting dominant spatial patterns from high-dimensional point cloud datasets. In structural dynamics applications, PCA enables the decomposition of complex vibration patterns into orthogonal principal components corresponding to fundamental vibration modes [5, 6]. For an undamped or lightly damped structure with uniform mass distribution, the principal directions converge to mode shape directions, with singular values indicating modal participation energy [4]. This transformation effectively converts geometric information into a friendly format to classical modal analysis techniques while preserving spatial richness.

Blind Source Separation (BSS) complements PCA by addressing the fundamental challenge of extracting independent modal responses from mixed observations without prior knowledge of source characteristics. BSS-based methods for modal identification treat modal coordinates as independent sources, enabling separation of closely-spaced modes and identification of operational deflection shapes under ambient excitation [7, 8]. The complexity pursuit algorithm has demonstrated particular effectiveness for estimating closely spaced and highly damped modes with minimal user supervision [9], making it suitable for processing high-dimensional point cloud data.

Current research gaps exist at the intersection of neural implicit representations and structural dynamics characterization. While NeRF technology has demonstrated success in static scene reconstruction, its application to dynamic structural systems remains largely unexplored. The temporal consistency requirements for modal analysis demand robust tracking of geometric features across multiple time instances, which challenges traditional NeRF formulations optimized for static scenes. Additionally, the applications of NeRF-based point cloud sequences for real-time structural monitoring requires systematic investigation.

The integration of NeRF-based 3D reconstruction with advanced signal processing techniques represents a significant advancement beyond existing point cloud-based modal analysis methods. Traditional approaches using ToF sensors are limited by frame rate constraints and depth resolution, restricting applications to structures with relatively low natural frequencies [4]. NeRF-based approaches, leveraging multi-view imagery with higher temporal resolution of high-speed cameras, could potentially overcome these limitations while providing enhanced spatial coverage and measurement density.

This paper presents a novel methodology that leverages NeRF-based 3D reconstruction for generating high-resolution point cloud representations of vibrating structures, followed by PCA-BSS processing for modal parameter extraction. The approach addresses fundamental limitations in current experimental modal analysis practices by combining the spatial richness of neural implicit representations with established blind identification techniques. The methodology extends recent advances in dynamic point cloud processing [4] by introducing NeRF-based reconstruction as a means to achieve superior temporal consistency and spatial resolution.

The proposed approach represents a paradigm shift from sparse sensor networks to dense, spatially distributed measurement systems capable of capturing intricate mode shape details that conventional accelerometer arrays cannot resolve. By integrating cutting-edge computer vision techniques with established structural dynamics theory, this work opens new frontiers in non-contact structural health monitoring with implications extending to robotics, aerospace, and biomedical applications.

While Neural Radiance Fields (NeRF) [3] have been successfully applied to the static 3D reconstruction of plants [10] and civil structures [11], their extension to dynamic scenes for quantitative modal analysis remains unexplored. This work presents, to the best of our knowledge, the first methodology to leverage the dense, view-consistent spatiotemporal reconstructions from dynamic NeRF for full-field operational modal analysis.

The application of Neural Radiance Fields (NeRF) to represent the spatiotemporal dynamics of plant vibration modes presents a novel methodology for the non-invasive extraction of quantitative physiological traits. This technique yields high-resolution, full-field quantitative data *in planta*, establishing a foundation for scalable phenotyping at the canopy level. In contrast to established but often cost-prohibitive and range-limited LiDAR systems, the proposed method leverages standard high-definition camera systems, enabling unprecedented data fidelity at a substantially reduced operational cost.

Our primary contributions include: (1) demonstration of NeRF applicability to dynamic structural systems for modal analysis applications, (2) development of processing workflows that balance computational efficiency with measurement fidelity, and (3) validation of the methodology against simulation based ground truth data. The results provide a foundation for advancing vision-based structural dynamics assessment and highlight the transformative potential of neural implicit representations in experimental mechanics.

Neural Radiance Fields

Neural radiance fields represent scenes as continuous volumetric functions, learned through differentiable rendering of multi-view images [12]. Unlike traditional photogrammetric approaches based on Structure-from-Motion (SfM) and Multi-View Stereo (MVS), NeRF-based methods demonstrate superior performance in handling complex geometries, reflective surfaces, and sparse viewpoints [13]. The extraction of dense point clouds from trained NeRF models provides geometrically consistent three-dimensional representations suitable for downstream structural analysis tasks [14].

MLP architecture

The NeRF architecture employs a multilayer perceptron (MLP) consisting of nine fully connected layers, each with 256 neurons and ReLU activation functions. A skip connection is incorporated into the fifth layer, where positional encoding [15] is introduced to enhance the network's ability to capture fine spatial details, as depicted in Figure 1. The ninth layer outputs the density parameter, σ. An additional tenth layer, with 128 neurons and ReLU activation, processes the remaining 256-dimensional feature vector. A final sigmoid-activated output layer predicts the emitted RGB radiance at position x, as observed along a ray with direction d [3].

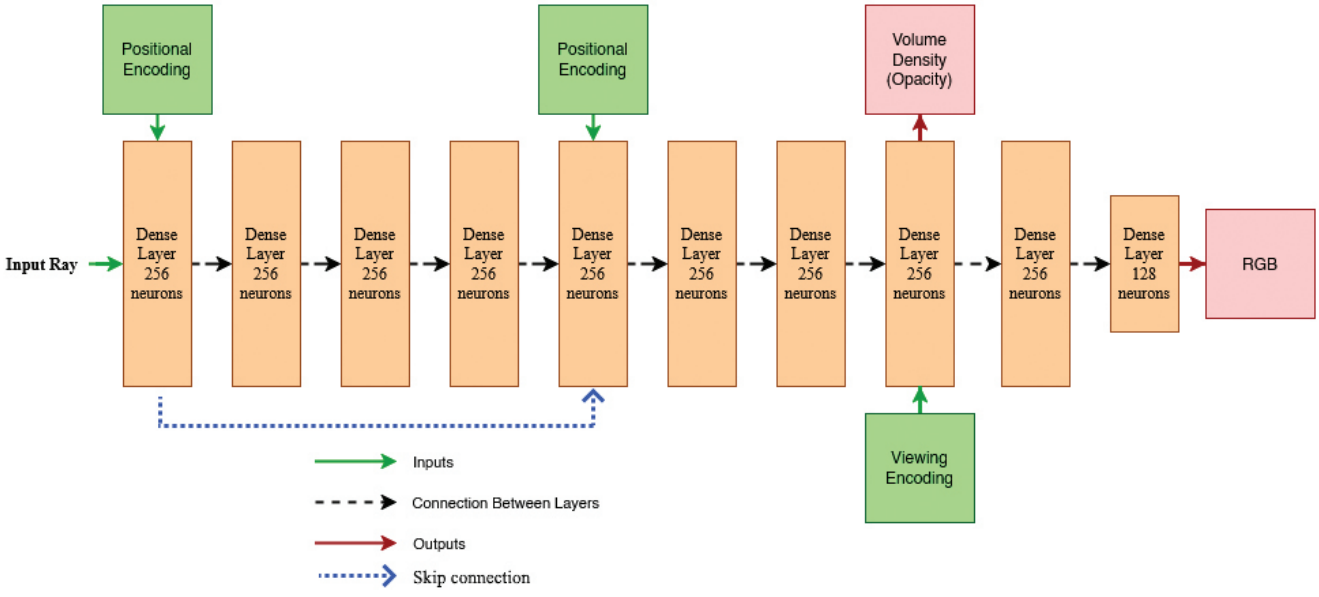


Fig. 1 Visualization of the NeRF's fully-connected network architecture.

Volume rendering

The author in [3] models the 5D neural radiance field to represent a scene as the volume density and directional emitted radiance at any point in space. The color of any ray passing through the scene is rendered using principles from classical volume rendering [16]. The volume density $\sigma(\mathbf{x})$ is interpreted as the differential probability of a ray terminating at an infinitesimal particle at location $\mathbf{x}$. Equation 1 depicts the expected color $C(\mathbf{r})$ of camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ with near and

far bounds t_n and t_f is:

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))c(\mathbf{r}(t), d)dt, \quad T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right). \tag{1}$$

The function $T(t)$ denotes the accumulated transmittance along the ray from t_n to t, i.e., the probability that the ray travels from t_n to t without hitting any other particle. Rendering a view from a continuous neural radiance field requires estimating this integral $C(\mathbf{r})$ for a camera ray traced through each pixel of the desired virtual camera.

This continuous integral can be numerically estimated using quadrature. Deterministic quadrature, which is typically used for rendering discretized voxel grids, would effectively limit the representation's resolution because the MLP would only be queried at a fixed discrete set of locations. Instead, a stratified sampling approach is used where its partitioned $[t_n, t_f]$ into N evenly-spaced bins and then draw one sample uniformly at random from within each bin, as depicted in Equation 2:

$$t_i \sim \mathcal{U}\left[t_n + \frac{i-1}{N}(t_f - t_n), t_n + \frac{i}{N}(t_f - t_n)\right]. \tag{2}$$

Although a discrete set of samples is used to estimate the integral, stratified sampling enables representing a continuous scene representation because it results in the MLP being evaluated at continuous positions over the course of optimization. These samples are used to estimate $C(\mathbf{r})$ with the quadrature rule discussed in the volume rendering review by Max [17]:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i\delta_i))c_i, \quad T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j\delta_j\right), \tag{3}$$

where $\delta_i = t_{i+1} - t_i$ is the distance between adjacent samples. This function for calculating $\hat{C}(\mathbf{r})$ from the set of (c_i, σ_i) values is trivially differentiable and reduces to traditional alpha compositing with alpha values $\alpha_i = 1 - \exp(-\sigma_i\delta_i)$.

Positional encoding

Positional encoding[1] is a method used to inject spatial information into neural networks, enabling them to effectively model high-frequency variations and complex spatial structures. It maps low-dimensional inputs, such as spatial coordinates (x, y, z) or viewing directions (θ, ϕ), into a higher-dimensional feature space using a series of sinusoidal functions. This technique was introduced in the context of Transformers [15] and has found applications in Large Language Models [18], audio processing [19] and graph-structured data [20]. It was also adapted for use in NeRFs [3], where capturing fine-grained spatial details is essential.

The positional encoding function is defined in Equation 4:

$$\text{PE}(p) = \left[\sin(2^k\pi p), \cos(2^k\pi p)\right]_{k=0}^{L-1} \tag{4}$$

where p represents the input coordinate (e.g., x, y, z), and L determines the number of frequency bands used in the encoding [15]. The canonical NeRF uses $L = 10$ frequencies for spatial coordinates (x, y, z) and $L = 4$ for viewing directions (θ, ϕ).

For a single input p, the positional encoding expands it into a $2L$-dimensional feature vector:

$$\text{PE}(p) \in R^{2L}. \tag{5}$$

For 3D spatial coordinates (x, y, z), the encoded representation is computed as:

$$\text{PE}(x, y, z) = [\text{PE}(x), \text{PE}(y), \text{PE}(z)], \tag{6}$$

resulting in a $6L$-dimensional feature vector for each 3D point. Similarly, for viewing directions (θ, ϕ) the encoded representation is:

$$\text{PE}(\theta, \phi) = [\text{PE}(\theta), \text{PE}(\phi)], \tag{7}$$

Positional encoding is crucial in overcoming the limitations of standard neural networks, which struggle to model high-frequency signals due to the inherent smoothness of activation functions like ReLU. By incorporating Fourier features, the network gains the capacity to represent fine details and sharp edges in data, significantly improving the quality of reconstructions and representations in tasks like 3D rendering and image synthesis.

[1] Positional encoding was inspired by Fourier feature mapping [3], a technique used to enable neural networks, especially MLPs, to better capture high-frequency signals in data, particularly in low-dimensional domains such as image and graphics processing.

Hierarchical volume sampling

NeRF's rendering strategy of densely evaluating the neural radiance field network at N query points along each camera ray is inefficient: free space and occluded regions that do not contribute to the rendered image are still sampled repeatedly. An hierarchical representation is proposed by [3] that increases rendering efficiency by allocating samples proportionally to their expected effect on the final rendering.

Instead of just using a single network to represent the scene, two networks are simultaneously optimized: one "coarse" $(N_c = 64)$ and one "fine" $(N_f = 192)$. First a set of N_c locations are sampled using stratified sampling, and evaluate the "coarse" network at these locations as described in equations (2) and (3). Given the output of this "coarse" network, it is then produced a more informed sampling of points along each ray where samples are biased towards the relevant parts of the volume, as depicted in Figure 2. To do this, first the alpha composited color is rewritten from the coarse network $\hat{C}_c(\mathbf{r})$ in equation (3) as a weighted sum of all sampled colors c_i along the ray, depicted in Equation 8:

$$\hat{C}_c(\mathbf{r}) = \sum_{i=1}^{N_c} w_i c_i, \qquad w_i = T_i(1 - \exp(-\sigma_i \delta_i)). \tag{8}$$

Normalizing these weights as $\hat{w}_i = w_i / \sum_{j=1}^{N_c} w_j$ produces a piecewise-constant Probability Density Function (PDF) along the ray. A second set of N_f locations is sampled from this distribution using inverse transform sampling, evaluate the "fine" network at the union of the first and second set of samples, and compute the final rendered color of the ray $\hat{C}_f(\mathbf{r})$ using the equation (3) but using all $N_c + N_f$ samples. This procedure allocates more samples to regions expected to contain visible content. This addresses a similar goal as importance sampling, but the sampled values are used as a non-uniform discretization of the whole integration domain rather than treating each sample as an independent probabilistic estimate of the entire integral.

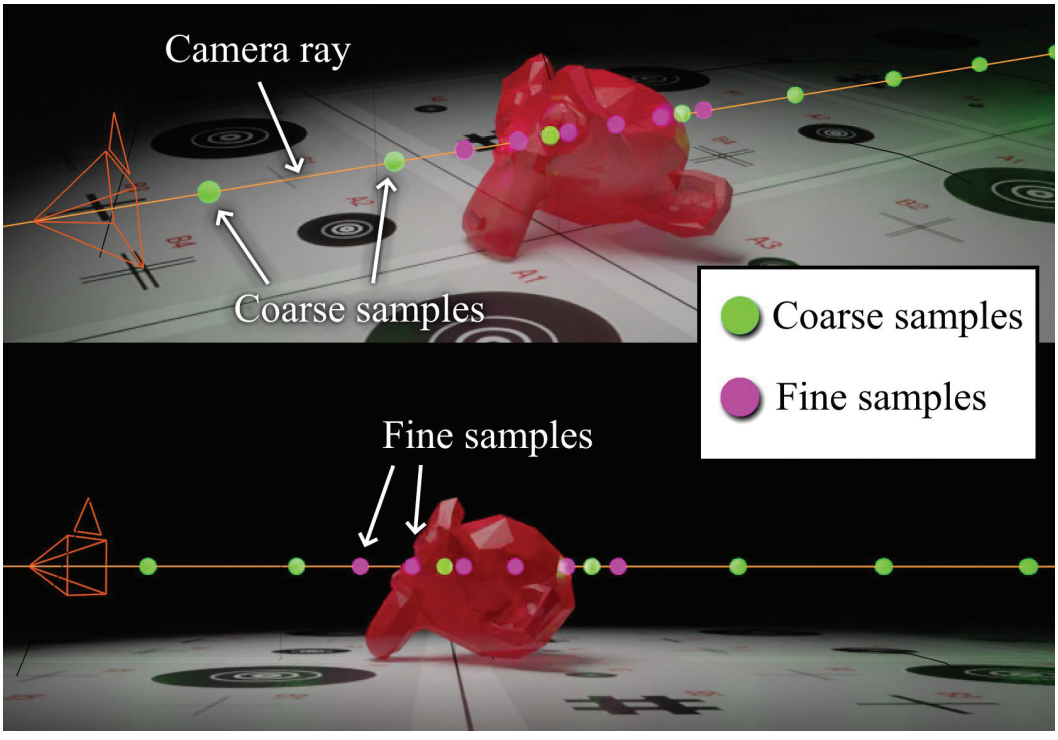


Fig. 2 Visual representation of NeRF's hierarchical sampling. First a set of coarse points (in green) are uniformly sampled along a particular ray using a coarse network to estimate general scene density. This initial density prediction is then used to form a probability distribution that guides the fine network to finely sample additional points (in pink) in regions that likely contain important details.

Loss function

During training of deep learning models, a loss function is used to optimize the model's parameters [21]. It measures the difference between the predicted and expected outputs of the model. The objective of training is to minimize this difference as the training iterations occur. Given the vast array of deep learning applications, different loss functions can be applied to different kind of problems. For instance, using 0-1 loss, perceptron loss, logarithmic loss, sigmoid cross entropy loss, and others for classification problems and Absolute loss, Huber Loss, Log-cosh loss and others in regression problems [22]. In image models, the loss function is computed by comparing the colors of a synthesized image with the ground-truth image [23]. Traditional image models rely on the L2 norm, also known as the Euclidean Norm, for image reconstruction [24].

In the context of NeRF training, the Loss function defined at Equation 9 was used to measure the difference between a predicted image $\hat{C}_p(r)$ and a ground truth image $C(r)$. The loss function of the NeRF is the total squared error between the rendered and true pixel colors for both the coarse $(N_c = 64)$ and fine $(N_f = 192)$ renderings:

$$L = \sum_{r \in R} \left[\|\hat{C}_c(r) - C(r)\|_2^2 + \|\hat{C}_f(r) - C(r)\|_2^2 \right]. \tag{9}$$

NeRFs are trained on batches of rays rather than entire images. R represents the set of rays in each training batch. For each ray r in a batch, $C(r)$ denotes the ground truth RGB color, while $\hat{C}_c(r)$ and $\hat{C}_f(r)$ represent the predicted RGB colors from the coarse and fine models, respectively. The training aims minimizing the difference between these predicted colors and the ground truth, ensuring that both coarse and fine models learn to accurately reconstruct the scene.

Point cloud extraction and export

Extracting explicit geometry information in the shape of a point cloud from an implicit volumetric representation of a scene from a trained NeRF model is done by sampling a batch of rays from different viewing angles. Using these rays, the NeRF model infers their RGB color and opacity for every sampled 3D coordinate point. Since the camera origins and directions of every ray are known, the distance/depth of every sampled point to the camera origin are also known from the camera extrinsics. The distance/depth from the camera's origin to the sample point is encoded into 3D coordinates in space.

$$P = \{x_i \mid \sigma(x_i) > \sigma_{threshold} \text{ for } x_i \in \mathbb{R}^3\}, \tag{10}$$

where $\sigma(x_i)$ is the predicted density at point (x_i) and $\sigma_{threshold}$ is a predefined opacity threshold used to extract surface points. Points below certain $\sigma_{threshold}$ (often 0.9) or outside a bounding box are discarded. The remaining valid points are gradually collected, with the process repeating until an arbitrary number of high quality points have been gathered. Finally, the accumulated 3D points and their color data can be exported into a standardized point cloud format for further processing. An advantage of implicit representations is the continuous nature of the reconstructed model. Therefore, given sufficient memory resources, point clouds containing an arbitrary number of points can be generated from any region of the model.

Motion Tracking by Virtual Lagrangian Sensors

Given a sequence of time-varying point cloud data, we define a densely populated grid of three-dimensional points (Figure 3). Each point in this grid is used as a measurement point for the motion tracking procedure. The goal is to generate virtual Lagrangian sensors by, allowing the efficient measurement of displacements at thousands of points across the structure's surface. In the case of Figure 3, we are able to densely monitor the structure using 99.990 virtual Lagrangian sensors for dynamic point cloud data with approximately 100.000 points. Note that the number of points in the point cloud data is not constant across time. As a result this modelling step is imperative to enable the measurement of motion in the same structure locations over time. Also, the number of virtual sensors can be arbitrarily chosen, and randomly sampled from the model. The concern is to sample the same points across time to simulate real sensors placed on the structure's surface as the points in the dynamic cloud are not suitable to directly use since the number of points change dynamically.

The dense number of sensing points is achieved by fitting a three-dimensional polynomial model over the point cloud data. A fixed set of points are sampled from the X- and Y-directions to measure displacements in the Z direction over time. Thus, motion is tracked on the same locations of the structure once the values for the Z-axis are interpolated by fitting the model. This process is repeated across time by fitting the three-dimensional model for each of the acquired point cloud data. Note that the model does not use any information regarding structure's properties or dimensions, instead, this unsupervised model is purely data-driven.

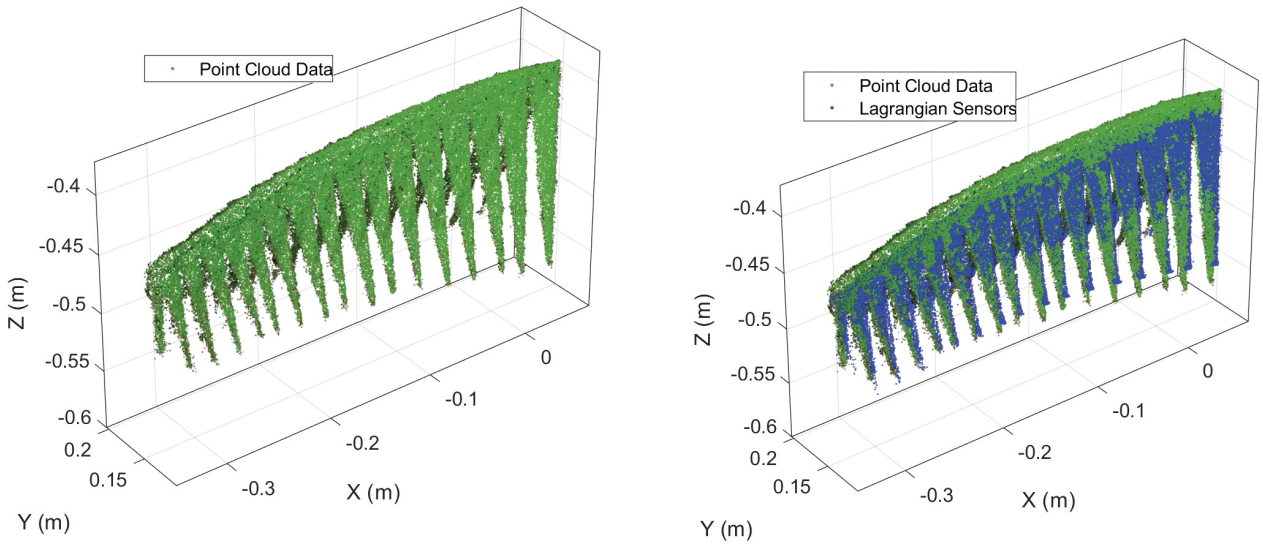


Fig. 3 Dynamic point cloud of a vibrating *Euterpe oleracea Mart.* (açaí palm) tree leaf (left) and a very dense grid of Lagrangian sensors on top of the point cloud data (right). In this case, we created 99.990 sensing points, which allows for full-field 3D monitoring for the entire leaf.

The displacement time series are the difference between the square root of the current position, z^i_{new} of the ith virtual sensor to its previous one, z^i_{old}, as in Equation (11), corresponding to a Lagrangian-type measurement once the values of the structure coordinates do not vary along time. Therefore, the end result of the motion tracking is a matrix $\Delta_z \in \mathbb{R}^{n\times m}$ with displacement measurements, where n is the number of dynamic point clouds (temporal dimension) and m the number of virtual Lagrangian sensors (spatial dimension).

$$\Delta^i_z = z^i_{new} - z^i_{old}. \tag{11}$$

Blind Identification of 3D Vibration Modes

Dimensionality Reduction

Once the vibration motion Δ_z is encoded into the motion time series from the dynamic point cloud data, we can directly apply conventional modal identification techniques over the collected time series. Here, we employ a recent developed class of algorithms for blind identification and visualization of vibration modes from video measurements [25, 26, 27].

Since our ultimate goal is to perform high resolution modal identification, it is desirable to use hundreds of virtual Lagrangian sensors. Thus, the first step for modal identification is to perform dimensionality reduction on the basis of Principal Component Analysis (PCA) over the motion time series. The goal is to factorize Δ_z in terms of two unknown matrices $\mathbf{U} \in \mathbb{R}^{m\times m}$ and $\mathbf{V} \in \mathbb{R}^{n\times n}$ such that $\mathbf{U}$ and $\mathbf{V}$ are the left- and right-singular vectors obtained by the eigenvalue decomposition (EVD) of the covariance matrix from Δ_z through Singular Value Decomposition (SVD) [28].

From a structural dynamics point of view [25, 26], it has been shown that for an undamped or very lightly damped structure, if its mass matrix is proportional to the identity matrix (i.e., uniform mass distribution), then the principal directions will converge to the mode shape direction [29, 27] with the corresponding singular values indicating their participating energy in the structural vibration responses, which means that the active structural modes are projected onto a very small number of r principal components

$$\eta \approx \Delta_z \mathbf{U}^T_r, \tag{12}$$

where $\mathbf{U}_r$ is the first $r(\ll n)$ columns of $\mathbf{U}$, and T denotes the transpose operation.

At this point, it is important to highlight the role being played by both $\mathbf{U}_r$ and η. As PCA is applied over the motion time series, $\eta \in \mathbb{R}^{n\times r}$ accounts for a temporal decomposition with each column η_i indicating the time pattern of a specific vibration mode. On the other hand, $\mathbf{U}_r$ incorporates a spatial decomposition with each row $\mathbf{U}_i$, when properly reshaped, resembling the corresponding vibration mode shape. Thus, when considering both η and $\mathbf{U}_r$ we have a 2D spatio-temporal decomposition.

Blind Identification of 3D Modal Parameters

Although η accounts for the vibration patterns extracted from the dynamic point cloud data, as mentioned in [29] and also highlighted by [27], the uniform mass distribution assumption is usually not satisfied, which makes each component η_i still a mixture of modes and can be approximated as a linear combination of the modal coordinates

$$\eta = \mathbf{q}\Psi = \sum_{i=1}^{r} q_i \psi_i, \tag{13}$$

where $\mathbf{q} \in \mathbb{R}^{n \times r}$ is the modal response matrix with q_i being the i-th modal coordinate, and $\Psi^{r \times r}$ is the mixing matrix. Using the relationship between the extracted components and the modal coordinates observed in [27] leads to

$$\Delta_z \approx \mathbf{q}\Psi\mathbf{U}_r = \mathbf{U}_r \sum_{i=1}^{r} q_i \psi_i = \sum_{i=1}^{r} q_i \langle \psi_i \mathbf{U}_r \rangle. \tag{14}$$

Comparing Equation (14) to Equation (13), high resolution mode shapes with resolution equal to the number of virtual sensors are given by

$$\Phi = \mathbf{U}_r \Psi^{-1}, \tag{15}$$

where $\Phi \in \mathbb{R}^{r \times m}$ is the mode shape matrix with each ϕ_i as the i-th mode shape. Similarly, modal coordinates can be estimated by

$$\mathbf{q} = \mathbf{U}_r \Psi. \tag{16}$$

The mixing matrix Ψ can be estimated by simply applying a BSS technique to solve the linear mixture problem stated in Equation (13). As previously demonstrated [25, 26], the complexity pursuit (CP) algorithm [30] can very efficiently estimate closely spaced and highly damped modes with small user supervision, and is therefore adopted here. Other modal parameters, such as resonant frequencies and damping ratios, can be further estimated using conventional Fourier transform and logarithmic decrement techniques.

Note that in this case the mode shapes in Φ are related to the motion in the Z-axis. Thus, in order to visualize a three-dimensional mode shape, the fixed X- and Y-coordinates for the virtual sensors can be used along with the mode shapes in the Z-axis for visualization of 3D modes.

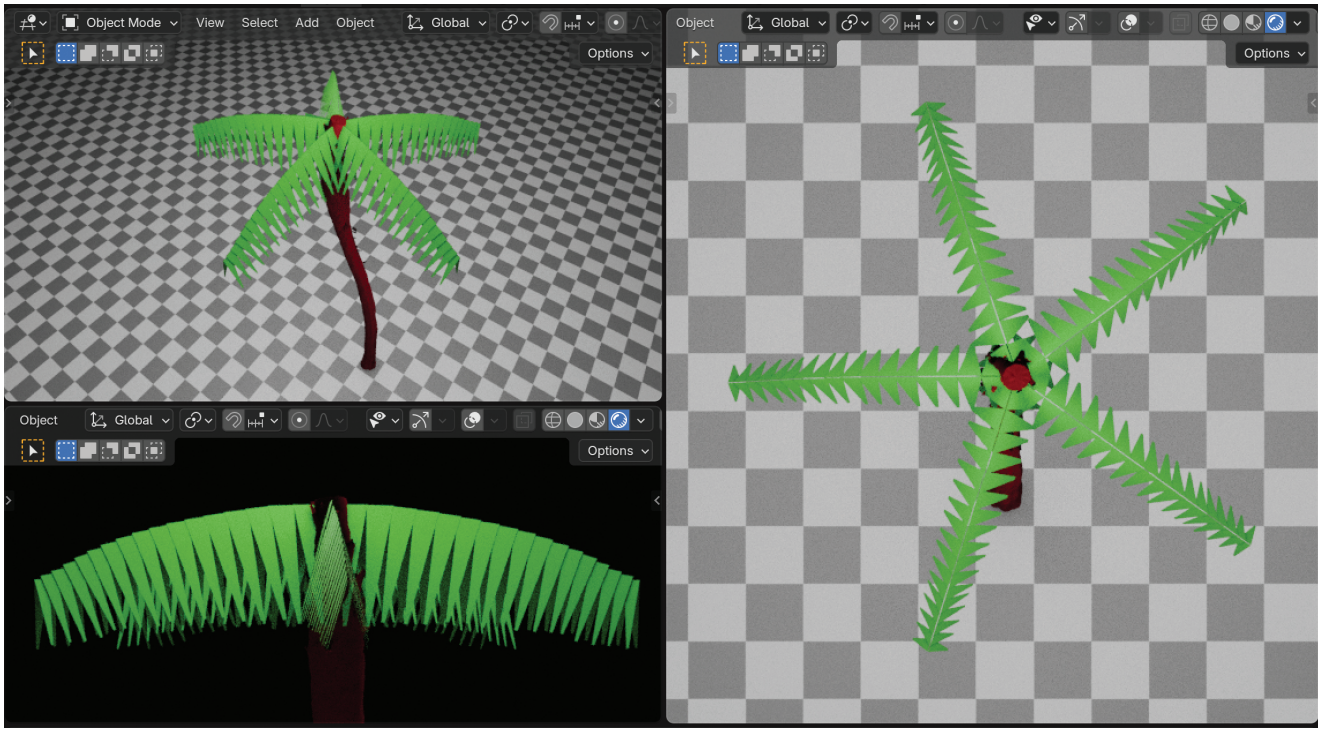


Fig. 4 Three-dimentional model of an *Euterpe oleracea Mart.* (açaí palm), modeled in Blender.

Modal Dynamics Visualization

To reconstruct each independent mode for visualization only, we can use Equation (13) and Equation (12) for a given mode i. First, multiplying the individual modal coordinate q_i in Equation (13) by a magnification factor α and the other ones by a decaying factor β we have

$$\hat{\eta}_i = (\alpha q_i)\psi_i + \beta \sum_{\substack{j=1 \\ j \neq i}}^{r} q_j \psi_j, \tag{17}$$

where $\hat{\eta}_i$ is the reconstructed temporal component after reconstruct the i-th mode. After that we can utilize the mode shape matrix Φ as our new set of spatial components, substitute into Equation (12) and then reconstruct a scaled version of the original measurement with the i-th mode enhanced by applying

$$\hat{\Delta}_z = \hat{\eta}_i \Phi. \tag{18}$$

The end result is a scaled version of the original video by a factor α while removing the other modes using a decaying term β. Other modes can be arbitrarily recovered for visualization purposes following the same procedure.

Estimating 3D Modal Parameters from NeRF-Based Dynamic Point Cloud Data

Data simulation with Blender

A simulation test was conducted to estimate modal parameters from a 3D *Euterpe oleracea Mart.* (açaí palm) tree measuring 2.59 meters long and 1.61 meters wide. The 2.59 meters long trunk geometry was generated using a cylindrical mesh with applied curvature modifiers, while the 0.98 meters long pinnate leaves were constructed through instanced meshes with simple extrusion and array operations to reproduce leaflet distribution. The scene is rendered with a basic PBR shading setup under HDRI illumination on a checkered reference plane, allowing evaluation of form, scale, and topology from multiple perspectives.

An açaí tree with 5 leaves and a branch was modeled. The leaves were modeled simulating z-axis vibration modes of 2hz (first bending mode), 5hz (second bending mode) and 10 hz (first torsional mode) while the branch had a single x-axis 1hz bending mode. 50 pictures from specific points were captured at each moment between 0 and 3 seconds (30 FPS). At each timestep, those 50 Full HD (1920p) pictures were used to train a nerf model which was then used to export an point cloud of that particular moment, totalizing 90 point clouds. Those point clouds were finally used as input for the PCA and BBS process in order to extract its vibration modes.

Similarly to [4] work, 3 principal components were extracted to estimate the first three structural modes. Figure 5 shows that there are 5 active eigenvalues. The presence of additional eigenvalues can be attributed to several inherent properties of the NeRF reconstruction process, such as:

1. Volumetric sampling artifacts arising from the discrete ray-casting procedure;
2. Temporal inconsistencies in point cloud density across timesteps, and/or;
3. Geometric reconstruction noise particularly pronounced at leaf boundaries and complex surface features.

Rather than representing spurious modes, these additional components capture the stochastic nature of the reconstruction process and provide a quantitative measure of method uncertainty.

Instead of fully training nerf models for each 90 timesteps 20k iterations, a transfer learning approach was done. The nerf model at timestep t=0 was initially trained for 17k iterations then for 3k more. The neural network weights for the initial 17k were used as starting weights for subsequent timestep models, each training an additional 3k iterations.

For the leaves, a 5x5 poly function was used to fit the points into an polinomial surface. For the branch, a cylinder fit was used.

Experimental results and discussion

Figure 6 and 7 shows the vibration modes extracted from the nerf-based point cloud models with PCA and BSS respectively. In this case, the resonant frequency estimated for the first bending mode was correctly identified at 2 Hz, while the second bending mode and the first torsional mode frequencies were identified as 5 Hz and 10 Hz.

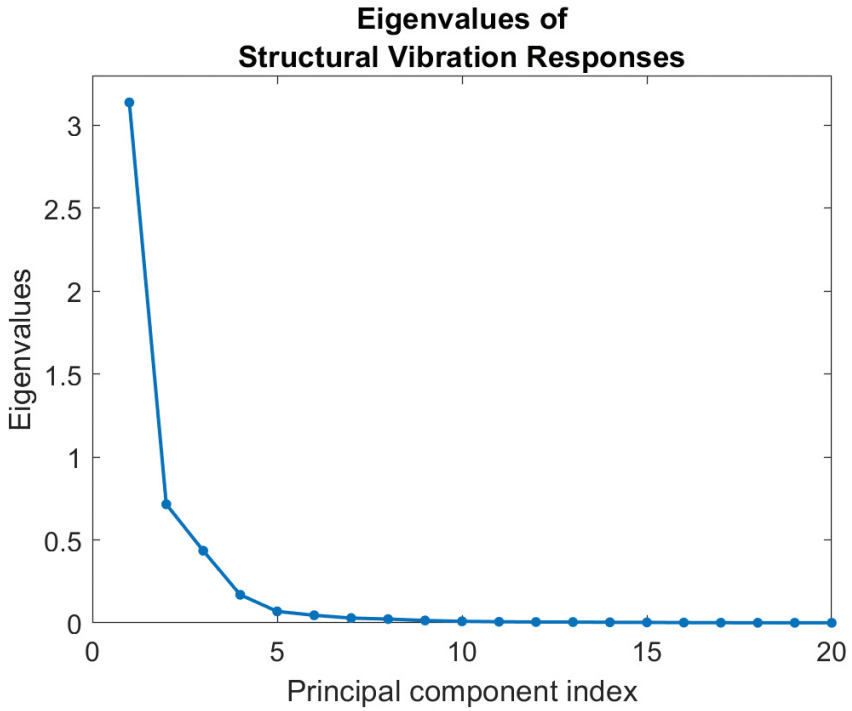


Fig. 5 The eigenvalue distribution of the covariance matrix of the structural responses from the proposed vision approach.

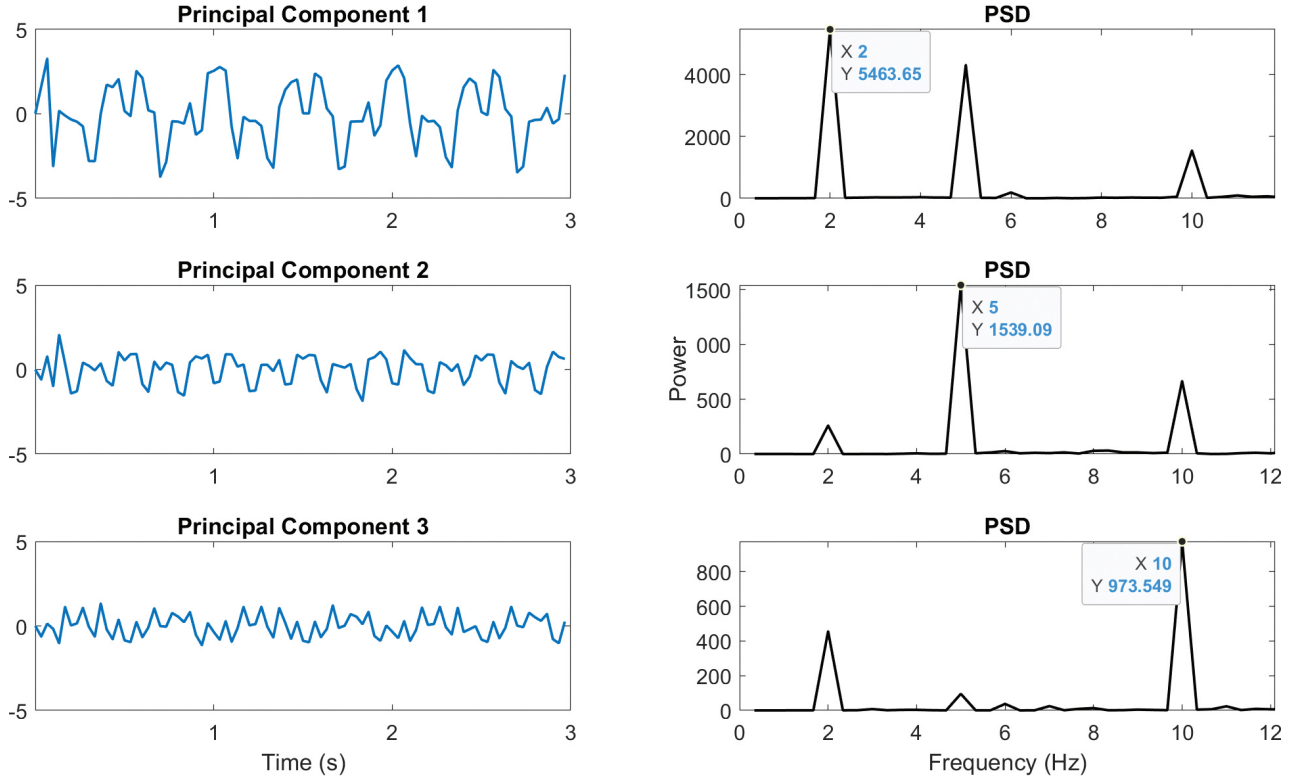


Fig. 6 The principal components and their power spectral density (PSD) of the structural responses from the proposed NeRF-based approach.

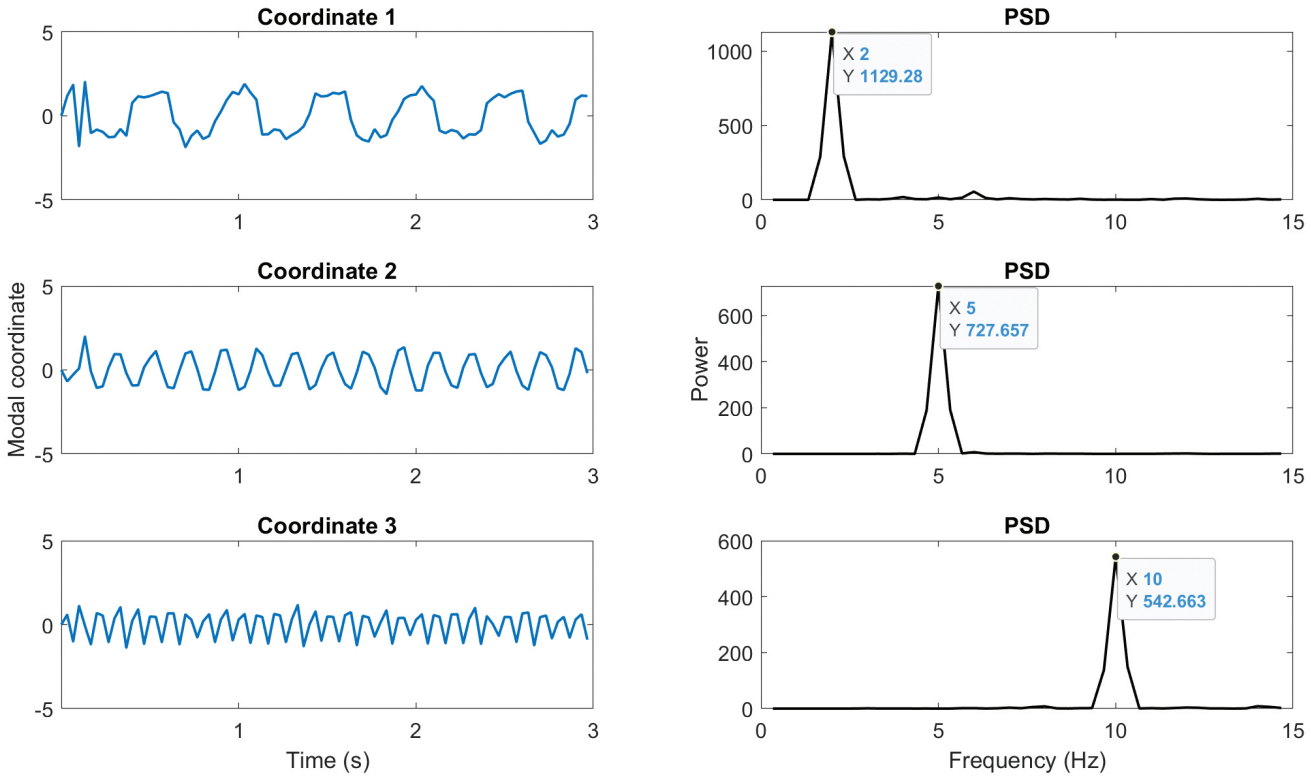


Fig. 7 Modal coordinates and their power spectral densities (PSD) estimated using the blind identification approach from the NeRF-based approach.

The mode static mode shapes are shown in Figure 8. For illustration purposes, several frames from the independent 3D mode shapes can be observed in Figure 9. For comparison purposes, the amplitude of the mode shapes were normalized to the same range. Note that those mode shapes are in perfect agreement with the expected simulated geometry and behavior for their respective modes.

Based on the experimental results, the proposed NeRF-based approach (as depicted in Figure 10) successfully extracted modal parameters from the simulated *Euterpe oleracea Mart.* palm structure. The method correctly identified all three target frequencies: the first bending mode at 2 Hz, second bending mode at 5 Hz, and first torsional mode at 10 Hz, demonstrating accurate frequency domain performance. Both PCA and BSS techniques yielded consistent results in extracting the three principal vibration modes from the 90 reconstructed point clouds. The recovered 3D mode shapes exhibited excellent agreement with the expected simulated geometry, validating the geometric accuracy of the approach. The presence of five active eigenvalues, rather than the expected three, was appropriately attributed to inherent reconstruction artifacts including volumetric sampling noise, temporal density variations, and geometric reconstruction uncertainties at complex features such as leaf boundaries.

The transfer learning strategy employed for NeRF training represents a computationally efficient approach, initializing subsequent timestep models with 17k pre-trained iterations and requiring only 3k additional iterations per frame. This methodology significantly reduces computational burden while maintaining reconstruction quality across the temporal sequence. The integration of polynomial surface fitting (5x5) for leaf structures demonstrates appropriate geometric parameterization for the distinct morphological components. The successful extraction of modal parameters from synthetically generated data establishes a proof-of-concept for vision-based structural health monitoring of vegetation, though real-world validation will require addressing additional challenges including environmental noise, varying illumination conditions, and the stochastic nature of natural wind excitation.

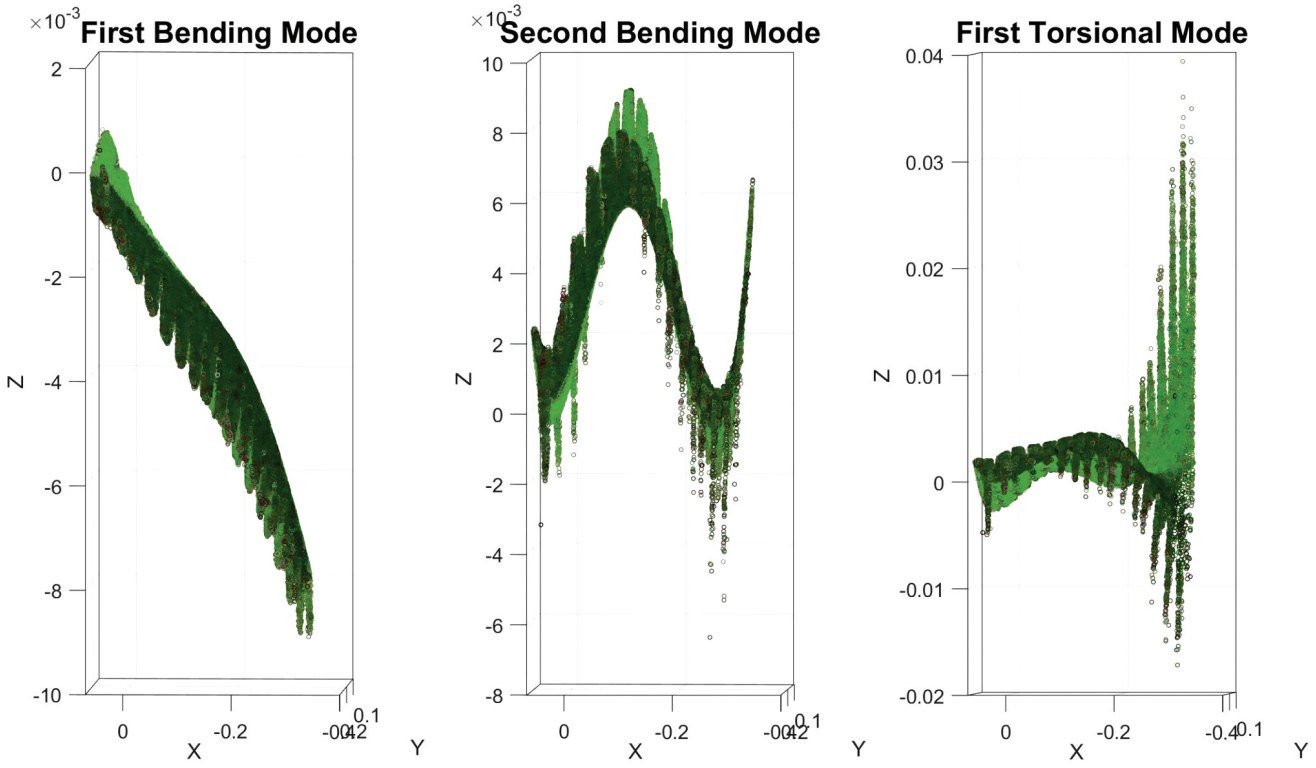


Fig. 8 3D mode shapes estimated from the point cloud data using the blind identification approach based on PCA and blind source separation.

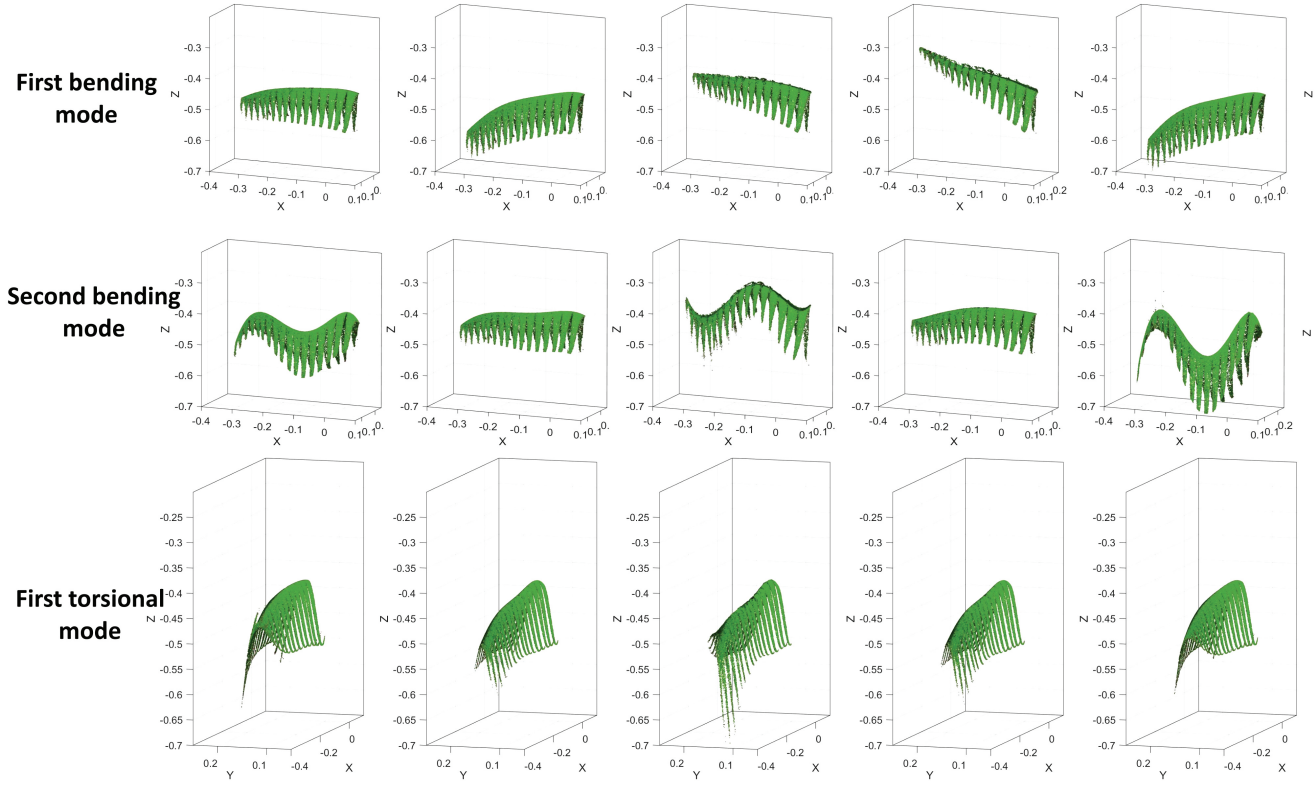


Fig. 9 A few temporal frames from the motion magnified 3D videos for each one of the vibration modes.

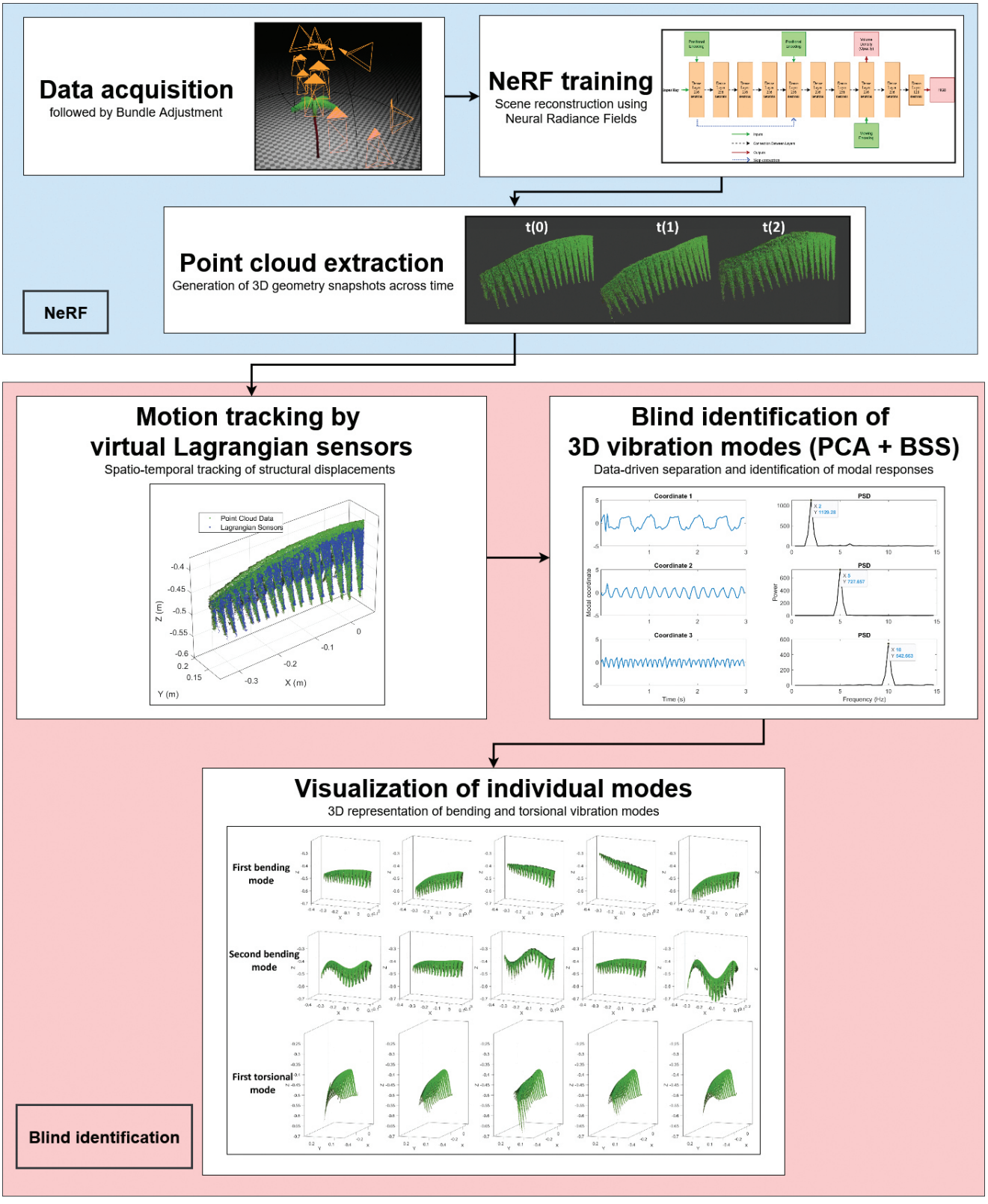


Fig. 10 We introduce the first attempt to process and extract 3D modal parameters from dynamic, NeRF-based, point cloud data. Initially, a set of pictures are obtained from a timestep of the structure. Then, these pictures are input into the NeRF pipeline where, after the training is finished, a point cloud of that timestep is obtained. This process is repeated until all models from the timeframe are trained and their point clouds extracted. Afterwards, blind identification of vibration modes is performed using PCA and BSS algorithms over motion time series estimated for each one of the coordinates of the structure. Finally, the mode shapes and modal coordinates are used to reconstruct the motion of the independent vibration modes.

Conclusion

The integration of machine learning-based 3D reconstruction with established modal analysis techniques represents a paradigm shift in structural dynamics characterization. This convergence enables unprecedented spatial and temporal resolution while maintaining the theoretical foundations of modal decomposition. The proposed NeRF-PCA-BSS pipeline establishes a foundation for next-generation structural health monitoring systems, combining the geometric fidelity of neural scene representations with the physical interpretability of modal analysis. Moreover, the proposed approach offers several technical advantages. The continuous representation inherent to NeRF enables high-resolution displacement tracking, critical for accurate modal parameter estimation. The combination of PCA and BSS provides robust performance in the presence of measurement noise and operational excitation. Furthermore, the framework naturally handles incomplete observations and occlusions through the learned scene priors embedded in the neural radiance field.

Acknowledgments This work was supported by CNPq (National Council for Scientific and Technological Development), Brazil (process no. 141761/2025-3).

References

1. Rainieri, C. "Perspectives of second-order blind identification for operational modal analysis of civil structures". *Shock Vib.*, 2014:1–9 (2014)ifnextchar.gobble.
2. Brincker, R. and Ventura, C. *Introduction to operational modal analysis*. John Wiley & Sons, Nashville, TN (2015)ifnextchar.gobble.
3. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., and Ng, R. "Nerf: representing scenes as neural radiance fields for view synthesis". volume 65, page 99–106, New York, NY, USA (2021)ifnextchar.gobble. Association for Computing Machinery.
4. Silva, M., Green, A., Morales, J., Meyerhofer, P., Yang, Y., Figueiredo, E., and Mascareñas, D. *Full-Field 3D Experimental Modal Analysis from Dynamic Point Clouds Measured Using a Time-of-Flight Imager*, pages 163–166 (2022)ifnextchar.gobble.
5. Poncelet, F., Kerschen, G., Golinval, J.C., and Marin, F. "System identification in structural dynamics using blind source separation techniques". In Davies, M.E., James, C.J., Abdallah, S.A., and Plumbley, M.D., editors, *Independent Component Analysis and Signal Separation*, pages 778–785, Berlin, Heidelberg (2007)ifnextchar.gobble. Springer Berlin Heidelberg.
6. Sadhu, A., Narasimhan, S., and Antoni, J. "A review of output-only structural mode identification literature employing blind source separation methods". *Mechanical Systems and Signal Processing*, 94:415–431 (2017)ifnextchar.gobble.
7. Zhang, T., Wang, C., and Zhang, Y. "Three-dimensional operational modal analysis based on self-iteration principal component extraction and direct matrix assembly". *Journal of Vibroengineering*, 19(8):6262–6276 (2017)ifnextchar.gobble.
8. Peeters, B. and Roeck, G.D. "Stochastic system identification for operational modal analysis: A review". *Journal of Dynamic Systems Measurement and Control-transactions of The Asme*, 123:659–667 (2001)ifnextchar.gobble.
9. Yang, Y. and Nagarajaiah, S. "Output-only modal identification by compressed sensing: Non-uniform low-rate random sampling". *Mechanical Systems and Signal Processing*, 56-57:15–34 (2015)ifnextchar.gobble.
10. Hu, K., Ying, W., Pan, Y., Kang, H., and Chen, C. "High-fidelity 3d reconstruction of plants using neural radiance fields". *Computers and Electronics in Agriculture*, 220:108848 (2024)ifnextchar.gobble. DOI:10.1016/j.compag.2024.108848.
11. Ribeiro, T., Brandão, F., Cardoso, V., Costa, J., Bittencourt, T., and Silva, M. "Neural radiance fields for low-cost, high-resolution 3d scanning of critical infrastructures" (2025)ifnextchar.gobble.
12. Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A., Mcallister, D., Kerr, J., and Kanazawa, A. "Nerfstudio: A modular framework for neural radiance field development" (2023)ifnextchar.gobble.
13. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., and Hedman, P. " Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields ". In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5460–5469, Los Alamitos, CA, USA (2022)ifnextchar.gobble. IEEE Computer Society.
14. Zhou, L., Wu, G., Zuo, Y., Chen, X., and Hu, H. "A comprehensive review of vision-based 3d reconstruction methods". *Sensors*, 24(7) (2024)ifnextchar.gobble.
15. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., and Polosukhin, I. "Attention is all you need" (2023)ifnextchar.gobble.
16. Kajiya, J. and von herzen, B. "Ray tracing volume densities". *ACM SIGGRAPH Computer Graphics*, 18:165–174 (1984)ifnextchar.gobble. DOI:10.1145/964965.808594.
17. Max, N. "Optical models for direct volume rendering". *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108 (1995)ifnextchar.gobble. DOI:10.1109/2945.468400.
18. Kazemnejad, A., Padhi, I., Ramamurthy, K.N., Das, P., and Reddy, S. "The impact of positional encoding on length generalization in transformers" (2023)ifnextchar.gobble.
19. Pepino, L., Riera, P., and Ferrer, L. "Study of positional encoding approaches for audio spectrogram transformers". In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, page 3713–3717. IEEE (2022)ifnextchar.gobble. DOI:10.1109/icassp43922.2022.9747742.
20. Cantürk, S., Liu, R., Lapointe-Gagné, O., Létourneau, V., Wolf, G., Beaini, D., and Rampášek, L. "Graph positional and structural encoder" (2024)ifnextchar.gobble.
21. Terven, J., Cordova-Esparza, D.M., Ramirez-Pedraza, A., Chavez-Urbiola, E.A., and Romero-Gonzalez, J.A. "Loss functions and metrics in deep learning" (2024)ifnextchar.gobble.

22. Wang, Q., Ma, Y., Zhao, K., and Tian, Y. "A comprehensive survey of loss functions in machine learning". *Annals of Data Science*, 9(2):187–212 (2020)ifnextchar.gobble. DOI:10.1007/s40745-020-00253-5.
23. Zhang, T., Zhu, D., Zhang, G., Shi, W., Liu, Y., Zhang, X., and Li, J. "Spatiotemporally enhanced photometric loss for self-supervised monocular depth estimation". In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE (2022)ifnextchar.gobble.
24. Zhao, H., Gallo, O., Frosio, I., and Kautz, J. "Loss functions for image restoration with neural networks". *IEEE Transactions on Computational Imaging*, 3(1):47–57 (2017)ifnextchar.gobble. DOI:10.1109/TCI.2016.2644865.
25. Yang, Y. and Nagarajaiah, S. "Blind modal identification of output-only structures in time-domain based on complexity pursuit". *Earthquake Engineering & Structural Dynamics*, 42(13):1885–1905 (2013)ifnextchar.gobble.
26. Yang, Y. and Nagarajaiah, S. "Structural damage identification via a combination of blind feature extraction and sparse representation classification". *Mechanical Systems and Signal Processing*, 45(1):1–23 (2014)ifnextchar.gobble.
27. Yang, Y., Dorn, C., Mancini, T., Talken, Z., Kenyon, G., Farrar, C., and Mascareñas, D. "Blind identification of full-field vibration modes from video measurements with phase-based video motion magnification". *Mechanical Systems and Signal Processing*, 85:567–590 (2017)ifnextchar.gobble.
28. Golub, G.H. and Van Loan, C.F. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, USA (1996)ifnextchar.gobble.
29. Feeny, B. and Kappagantu, R. "On the physical interpretation of proper orthogonal modes in vibrations". *Journal of Sound and Vibration*, 211(4):607–616 (1998)ifnextchar.gobble.
30. Stone, J.V. "Blind source separation using temporal predictability". *Neural Computation*, 13(7):1559–1574 (2001)ifnextchar.gobble.

Chapter 2
Investigation on the identification of Spatially-Varying Elastic and Damping Properties in Beams and Plates Using Physics-Informed Neural Networks

Morvan Ouisse, Maxime Auger, João Gabriel Oliveira Aveiro, Roberta Tittarelli, Valentin Calisti, Rafael Teloli, Emmanuel Ramasso, Patrice Le Moal

Abstract This work presents a Physics-Informed Neural Network (PINN) approach for the identification of spatially-varying elastic and damping properties in structures subjected to bending. The study focuses on beams and plates made of non-homogeneous materials, where conventional identification techniques often struggle to accurately resolve local variations in mechanical properties due to limited data or the need for strong prior assumptions.

The proposed method investigates the ability of PINNs to embed the governing partial differential equations of structural dynamics directly into the training process of neural networks. By incorporating the equations of motion for bending, including both stiffness and damping contributions, into the loss function, the network learns not only to fit the observed displacement data, but also to satisfy the underlying physics. This allows the identification of spatial distributions of the Young's modulus and damping coefficients without requiring explicit meshing or parameter regularization.

The approach is validated on simulated and experimental test cases involving non-uniform beams and plates, for steady-state flexural responses. Results discuss the ability of the PINN framework of accurately reconstructing complex spatial profiles of material properties, even with sparse and noisy measurements. The method is checked in terms of robustness and generalizability, to investigate its capability to become a promising tool for non-destructive testing and material characterization in advanced mechanical systems.

Keywords PINN · identification · damping

Introduction

Decision-making in engineering requires models that balance performance, validity and cost-effectiveness. Physics-based simulations, often formulated with partial differential equations (PDEs), serve as virtual prototyping tools to analyze complex structures. When analytical solutions are unavailable, numerical methods such as finite elements (FE) are employed, though they face challenges related to boundary conditions, meshing, and computational cost.

In parallel, data-driven methods have emerged, exploiting machine learning to address tasks from regression to structural monitoring. Despite their flexibility and universal approximation capabilities, such models often suffer from lack of interpretability, instability, and poor extrapolation [1, 2]. This motivates approaches that combine data with physical principles.

Physics-informed machine learning addresses this need by embedding physical laws into the training process [3]. Among these, Physics-informed neural networks (PINNs) [4] have gained traction, with applications in continuum mechanics [5], thermochemical processes [6], and fluid dynamics [7]. PINNs incorporate PDEs and conservation laws directly into the loss

Morvan Ouisse · Maxime Auger · João Gabriel Oliveira Aveiro · Roberta Tittarelli · Valentin Calisti · Rafael Teloli · Emmanuel Ramasso · Patrice Le Moal
Université Marie et Louis Pasteur, SUPMICROTECH, CNRS
Institut FEMTO-ST
F-25 000 Besançon, France
e-mail: morvan.ouisse@femto-st.fr; maxime.auger@femto-st.fr; joao.oliveira_aveiro@ens2m.org; roberta.tittarelli@femto-st.fr; valentin.calisti@femto-st.fr; rafael.teloli@femto-st.fr; emmanuel.ramasso@femto-st.fr; patrice.lemoal@femto-st.fr

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

B. J. Dilworth et al. (eds.), *Topics in Modal Analysis & Parameter Identification, Vol. 9*, Conference Proceedings of the Society for Experimental Mechanics Series, https://doi.org/10.13052/rp-9788743814207

function, bridging the gap between data and physics. Recent works have applied them to solid mechanics [8], viscoelasticity [9], and dynamical systems [10, 11].

However, their use in inverse problems for structural dynamics remains limited, particularly where damping and frequency-domain analysis are central [12, 13]. Existing studies on beam-like structures [14, 15, 16] do not validate with experimental data and often overlook damping, despite the sensitivity of high-order PDEs to noise [17, 18]. The work presented in this paper is an extension of the work published in a recent paper [19], in which a PINN-based framework has been introduced to identify beam structures, estimating both transverse displacement and complex elastic modulus directly in the frequency domain. The contribution is based on the identification of spatially-varying elastic and damping properties in beams and plates.

Physics Informed Neural Networks (PINNs)

PINNs extend classical neural networks by embedding physical constraints into their training. A fully connected feedforward neural network with L layers and parameters $\theta = \{W^{(\ell)}, b^{(\ell)}\}_{\ell=1}^{L}$ approximates a target function $y(\mathbf{x})$ by

$$\hat{y}(\mathbf{x};\theta) = \mathcal{H}^{(L)} \circ \cdots \circ \mathcal{H}^{(1)}(\mathbf{x}), \tag{1}$$

where each layer mapping is defined as

$$\mathcal{H}^{(\ell)}(\mathbf{z}) = \sigma\left(W^{(\ell)}\mathbf{z} + b^{(\ell)}\right), \tag{2}$$

with σ a nonlinear activation function.

In a purely data-driven setting, training minimizes the discrete squared L^2-norm

$$\mathcal{R}_d(\theta) = \frac{1}{N}\sum_{n=1}^{N} \left|\hat{y}(\mathbf{x}_n;\theta) - y(\mathbf{x}_n)\right|^2. \tag{3}$$

Here, the set $\mathbf{x}_n$ denotes the observation points, i.e., the locations where data $y(\mathbf{x}_n)$ is available. PINNs enrich this formulation by incorporating the residual of governing PDEs. For example, consider

$$\mathcal{D}^{\alpha}(y(\mathbf{x});\lambda) = f(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{4}$$

where $\mathcal{D}^{\alpha}$ is a differential operator parameterized by λ. This leads to the physics-informed loss

$$\mathcal{R}_f(\theta) = \frac{1}{M}\sum_{m=1}^{M} \left|\mathcal{D}^{\alpha}(\hat{y}(\mathbf{x}_m;\theta);\lambda) - f(\mathbf{x}_m)\right|^2. \tag{5}$$

where the set $\mathbf{x}_m$ denotes the collocation points, i.e., the locations in the domain where the differential equation is enforced.

The total loss $\mathcal{L}$ is then defined as a weighted combination of the two contributions:

$$\mathcal{L}(\theta) = \gamma\,\mathcal{R}_d(\theta) + \beta\,\mathcal{R}_f(\theta). \tag{6}$$

For inverse problems, unknown parameters λ can also be approximated by a neural network $\hat{\lambda}(\mathbf{x};\xi)$. This allows PINNs to jointly identify both the solution $y(\mathbf{x})$ and spatially varying properties $\lambda(\mathbf{x})$, making them particularly useful in structural dynamics for tasks such as material property mapping or damage detection.

Euler-Bernoulli Beam case

Description of the framework and its adaptation to the Euler-Bernoulli beam case

This section presents the PDE formulation of an Euler-Bernoulli model with a spatially varying elastic modulus and with a complex elastic modulus. The motion of an Euler-Bernoulli beam in permanent harmonic regime at angular frequency ω is driven by

$$\frac{\partial^2}{\partial x^2}\left(E(x)I\frac{\partial^2 w(x,\omega)}{\partial x^2}\right) - \rho S\omega^2 w(x,\omega) = f(x,\omega), \tag{7}$$

where $w = w(x, \omega)$ is the transversal deflection of the beam; I and S are the geometrical properties, corresponding to the flexural inertia $\frac{b \cdot h^3}{12}$ and the area of the cross section $h \times b$ of the beam, respectively; $E = E(x)$ and ρ [kg/m^3] are the material properties, representing the elastic modulus and density, respectively; $f = f(x, \omega)$ is the applied loading. The PDE (7) is a strong formulation, meaning the equation must be satisfied over the entire domain of the structure. This study focuses on a specific region of the beam where no external force is applied, to recover the parameters of the equation.

As the elastic modulus is non-uniform and differentiable along with the length x of the beam, and the material is linear viscoelastic, one has $E(x) = E_r(x) + jE_i(x)$, and the transversal deflection of the beam writes $w(x, \omega) = w_r(x, \omega) + jw_i(x, \omega)$. To simplify the notation, the dependence of the displacement and the complex elastic modulus on (x, ω) are omitted from now on. The loss factor η, which characterizes the material damping to be identified, is

$$\eta = \frac{E_i}{E_r}. \tag{8}$$

The problem-solving framework is presented in Figure 1. It aims at identifying the deflection of the beam and the non-constant elastic modulus, both of which are complex quantities. It includes four independent neural networks, i.e. $\hat{w}_r \approx w_r$, $\hat{w}_i \approx w_i$, $\hat{E}_r \approx E_r$ and $\hat{E}_i \approx E_i$, with parameters $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$, respectively. For simplifying notation, $\mathcal{W} := (\hat{w}_r(\cdot; \boldsymbol{\theta}_1), \hat{w}_i(\cdot; \boldsymbol{\theta}_2))$ and $\mathcal{E} := (\hat{E}_r(\cdot; \boldsymbol{\xi}_1), \hat{E}_i(\cdot; \boldsymbol{\xi}_2))$. Therefore, the physics-informed loss function is composed of two parts [19], one related to the real terms $\mathcal{R}_f^{(\text{real})}(\mathcal{W}, \mathcal{E})$ and the other involving the imaginary terms $\mathcal{R}_f^{(\text{imag})}(\mathcal{W}, \mathcal{E})$, see Fig. 1 for the expressions.

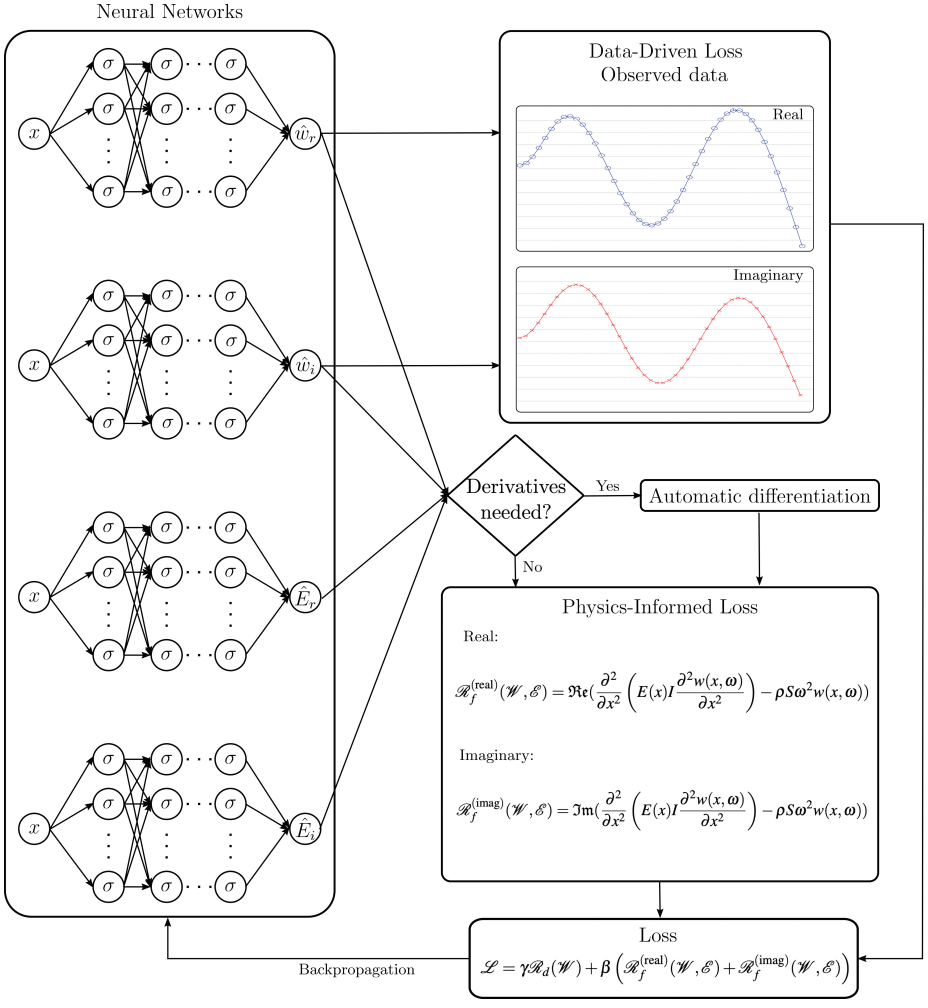


Fig. 1 PINN framework

In a similar manner, residual terms related to data observations can also be defined concerning real and imaginary components, resulting in

$$\mathcal{R}_d(\mathcal{W}) = \frac{1}{N_x}\sum_{n=1}^{N_x}\left|\hat{w}_{r,n} - w_{r,n}\right|^2 + \frac{1}{N_x}\sum_{n=1}^{N_x}\left|\hat{w}_{i,n} - w_{i,n}\right|^2, \tag{9}$$

where $\hat{w}_{\cdot,n}$ and $w_{\cdot,n}$ are the displacements evaluated at $\boldsymbol{x}_n$, and N_x corresponds to the number of points used for displacement measurements along the length of the beam. Finally, the loss function is given by

$$\mathcal{L} = \gamma\mathcal{R}_d(\mathcal{W}) + \beta\left(\mathcal{R}_f^{(\text{real})}(\mathcal{W},\mathcal{E}) + \mathcal{R}_f^{(\text{imag})}(\mathcal{W},\mathcal{E})\right), \tag{10}$$

where (γ,β) are determined heuristically. $\gamma = 1$ and $\beta = 10^{-2}$ are used in the following study.

Within the Euler Bernoulli theory, the strong form PDE involves the fourth spatial derivative of the transverse displacement. Spectrally, the operator scales like k^4, k being the wavenumber: small oscillations in the solution displacement field stemming from noise, overfitting, or numerical approximations are strongly magnified in the residual. As a consequence, residual minimization can become stiff and a naive optimizer may reduce the loss by injecting short wavelength oscillations into the inferred modulus. Our design avoids numerical differentiation of measurements, adopts smooth low variance representations for material fields, and relies on automatic differentiation for high order terms, which collectively mitigates this issue without changing the training pipeline.

Application to the identification of the properties of a beam equipped with a viscoelastic patch

Additionally to the cases already presented in [19], the efficiency of the method is tested on a case with radical properties change: a homogeneous steel beam partially covered with a viscoelastic patch. The 1-meter long cantilever beam (section 20×2 mm^2) has a loss factor of 0.5% and is excited by a point force located close to the clamp. A thin 10 cm-long viscoelastic patch is glued on the top surface of the beam, 70 cm away from the clamp. Its effect on the stiffness is negligible while it increases drastically the damping in the patch area. The RKU model [20] provides an equivalent loss factor of 0.5% in the uncovered zone suddenly growing to 50% in the viscoelastic patch area. From the knowledge of the displacement on a finite number of points along the beam, the network is expected to recover the displacement field on the zone covered by the measurements and identify the spatially varying elastic and damping properties. In the case illustrated below, the measurement zone covers half of the beam starting from 0.5 m away from the clamp, with 100 points regularly distributed on this area. These data are obtained from a finite element simulation.

This difficult case requires a huge number of iterations in the learning phase of the PINN to converge. Therefore, a variant is introduced to speed up the process, based on the assumption that the properties to identify are constant by parts on the beam, while the location of their changes remain unknown. To this end, this *a priori* information on the shape of spatial parameters is provided by introducing piecewise smooth plateaus separated by localized transitions (e.g., material patches, interfaces). Encoding this structure directly reduces the effective spatial bandwidth of admissible solutions and curbs spurious oscillations induced by high order derivatives. This is implemented using an arbitrary number of adjacent plateaus with smooth transitions build from sigmoids functions. Their strictly positive widths are enforced by parameterizing unconstrained raw gaps via softplus functions so that the intervals partition the beam length without overlap. The parameters fields $E_r(x)$ and $E_i(x)$ are realized by two single hidden layer networks whose hidden units are sigmoids positioned at the plateaus edges.

This *a priori* provides additional knowledge to the networks, while both the number of plateaus and their locations remain unknow and must be identified. To illustrate the efficiency of the method, 6 bounded plateaus with learnable edges and amplitudes are trained, while only 3 would be necessary to recover the exact solution. The results in terms of displacements are shown in Figure 2. The model preserves an accurate fit to complex displacements across the measured span, yielding smooth predictions between sensors thanks to the physical residual.

The architecture of the network is based on the one presented in [19], except the two parts related to the estimation of the mechanical properties whose networks are replaced by the plateaus-based approach described above.

The identified moduli show plateau–like behavior with physically plausible transitions. The loss modulus E_i is non negative and localized within the patch; E_r recovers steel levels outside, as illustrated in Figure 3.

This illustrates the ability of the framework to handle complex cases with radical changes in the mechanical properties, in particular the damping, whose estimation remains a challenge in structural dynamics. The methods requires the measurement of the displacement over a part of the structure, without the need to measure the input force and without the requirement to know the boundary conditions. This choice to localize informations in the training explains the behavior of the predicted modulus at the boundary of interval $[0.5, 1.0]$ in Figure 3, that is not to be taken into account. The next section aims at extending the approach to plates.

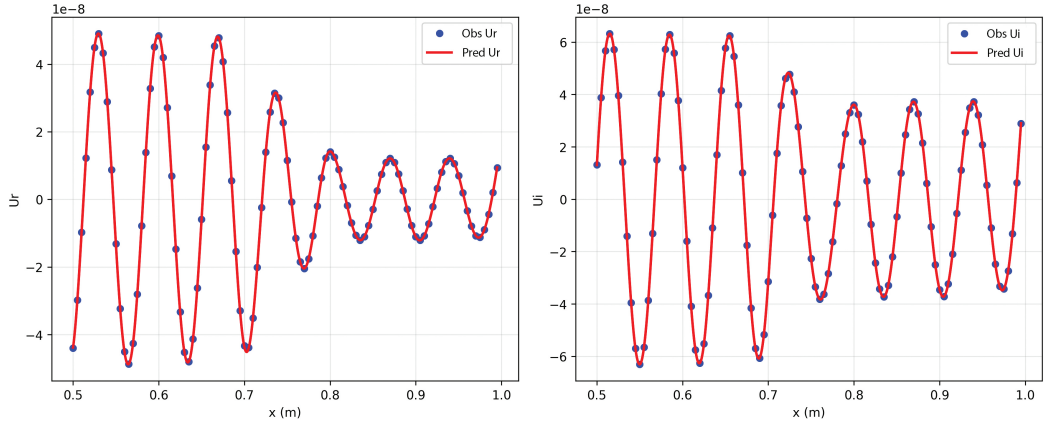


Fig. 2 Observed vs. predicted complex displacements at 4 kHz over the measured region for the 6 plateaus model

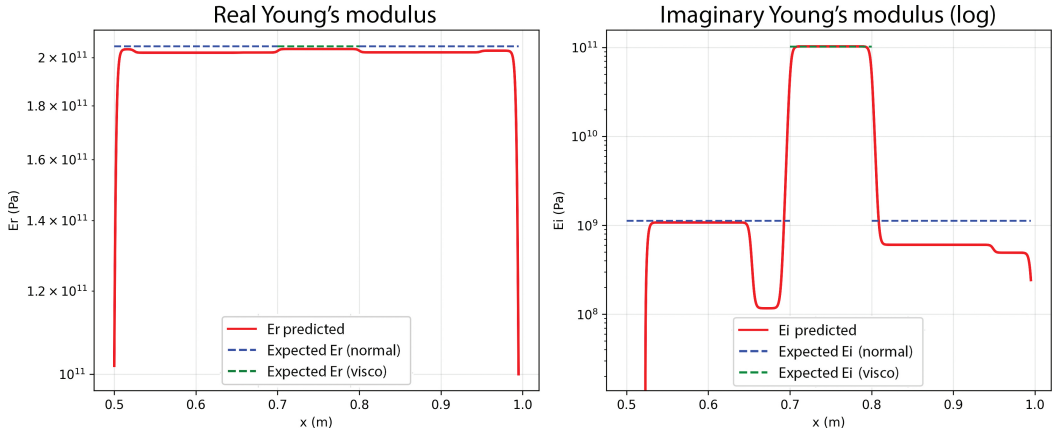


Fig. 3 Results with 6 learnable bounded plateaus: predicted real (left) and imaginary (right) parts of the modulus. Clean plateaus and physically plausible transitions are obtained.

Kirchhoff plate case

The framework introduced in [19] can be extended to any mechanical system whose movement is driven by a PDE. This section describes the basics of its application to a Kirchhoff plate, whose behaviour under harmonic motion at angular frequency ω is driven by [21]

$$-\rho h\omega^2 w + \frac{\partial^2}{\partial x^2}\left(D(x,y)\,\frac{\partial^2 w}{\partial x^2}\right) + 2\,\frac{\partial^2}{\partial x \partial y}\left(D(x,y)\,\frac{\partial^2 w}{\partial x \partial y}\right) + \frac{\partial^2}{\partial y^2}\left(D(x,y)\,\frac{\partial^2 w}{\partial y^2}\right) = q(x,y), \tag{11}$$

with ρ the density, h the thickness, $w(x,y)$ the transverse displacement, $q(x,y)$ the distributed transverse load and $D(x,y) = \frac{E(x,y)h^3}{12(1-\nu^2)}$ the bending modulus with $E(x,y)$ is the elastic modulus and ν the Poisson ratio. The split in real and imaginary parts required to define the physical losses leads to

$$-\rho h\omega^2 w_r + \partial^2_{xx}(D_r\, w_{r,xx} - D_i\, w_{i,xx}) + 2\,\partial^2_{xy}(D_r\, w_{r,xy} - D_i\, w_{i,xy}) + \partial^2_{yy}(D_r\, w_{r,yy} - D_i\, w_{i,yy}) = q_r, \tag{12}$$

$$-\rho h\omega^2 w_i + \partial^2_{xx}(D_r\, w_{i,xx} + D_i\, w_{r,xx}) + 2\,\partial^2_{xy}(D_r\, w_{i,xy} + D_i\, w_{r,xy}) + \partial^2_{yy}(D_r\, w_{i,yy} + D_i\, w_{r,yy}) = q_i, \tag{13}$$

with $\partial^2_{\alpha\beta} = \frac{\partial^2}{\partial\alpha\beta}$ and $w_{\gamma,\alpha\beta} = \frac{\partial w^2_\gamma}{\partial\alpha\beta}$.

To validate the proposed framework, a benchmark numerical experiment is first conducted using an aluminum plate (0.7 m $\times$ 0.8 m $\times$ 0.002 m) with homogeneous properties ($\eta = 0.2\%$). Four neural networks are employed: two for the displacement (one for the real part and one for the imaginary part) and two for the real and imaginary parts of the complex modulus E. All networks use an architecture with 5 hidden layers and 40 neurons per layer, with $tanh$ as the activation

function. 900 collocation points are used for the evaluation of the PDE, while 64 observation points are used to feed the network with virtual measurements. A maximum learning rate of 0.001 and 0.0001 was used for the displacement and modulus networks, respectively.

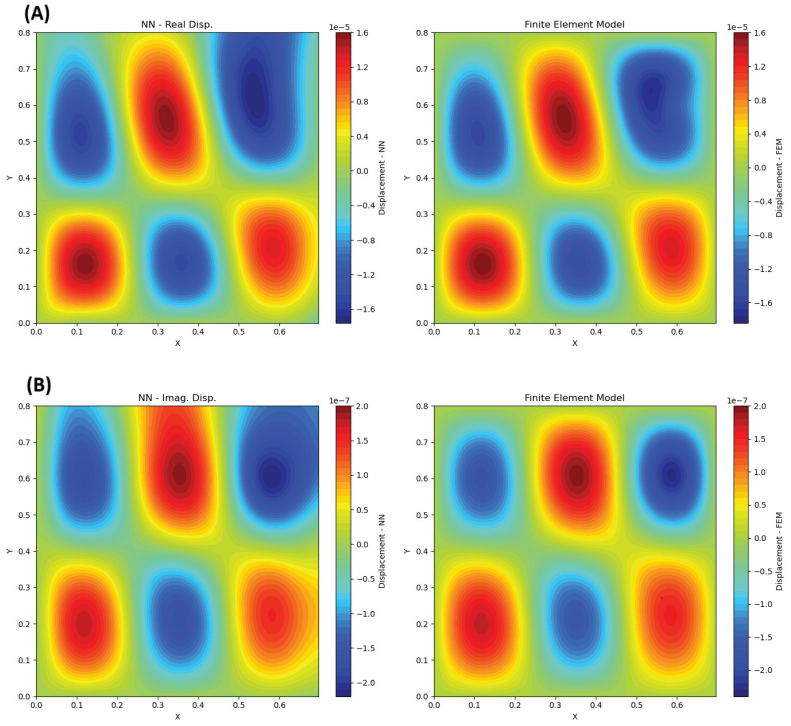


Fig. 4 Real (A) and imaginary (B) components of displacements at 130 Hz over the measured region for the homogeneous plate: PINN estimates (left) and FEM reference (right)

As illustrated in figure 4, the network provides a reasonably accurate estimation of both the real and complex components of the displacement field, even with a relatively limited number of observation points (64). The most pronounced deviations are observed near the boundary regions, which is consistent with the fact that the boundary conditions were not explicitly included in the network's prior knowledge.

The figure 5 shows the spatial distributions of the identified mechanical properties. The elastic modulus is estimated to be nearly constant over the entire surface, with variations on the order of one percent. The loss factor is predicted with good average accuracy (0.21% instead of 0.2%), although more pronounced spatial fluctuations are observed (ranging approximately from 0.1% to 0.3%). While these results are already encouraging, further improvements are expected through additional refinement of the network architecture and its hyperparameters.

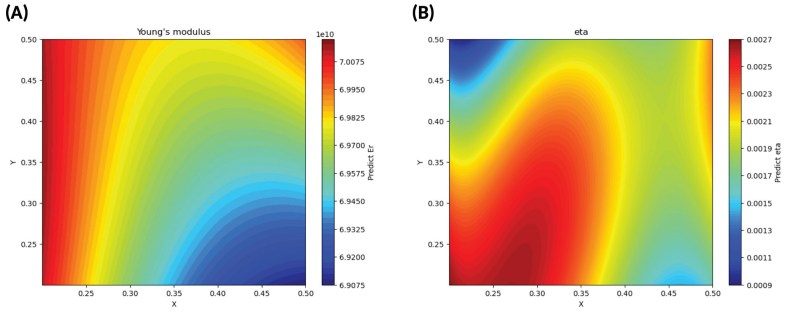


Fig. 5 Identified properties of the homogeneous plate: elastic modulus (A) and loss factor (B)

Finally, the case of a non-homogeneous conservative plate ($\eta = 0$) is investigated. A spatial variation of the elastic modulus is introduced in the vicinity of the point (0.35, 0.35), where this local reduction simulates, for instance, a defect

to be identified in a structural health monitoring (SHM) framework. The displacement field at an arbitrary frequency is computed, and 64 measurement points are employed for network training.

The displacement field is shown in figure 6. Despite its spatial complexity, it is remarkably well estimated by the network. As expected, the estimation is least accurate in the vicinity of the boundary conditions.

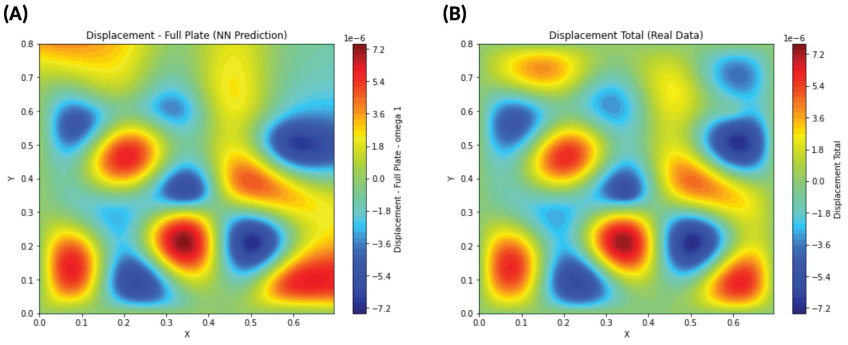


Fig. 6 Displacement at 295 Hz over the measured region for the non-homogeneous plate: PINN estimates (A) and FEM reference (B)

Figure 7 presents the elastic modulus identified by the network alongside the reference values. While the identification of the damaged region is not perfect, its location is correctly captured. These results demonstrate that the proposed approach offers promising perspectives for structural health monitoring.

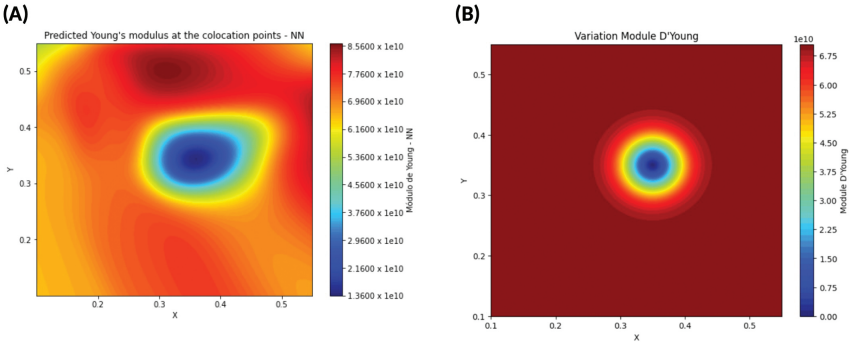


Fig. 7 Identified elastic modulus of the non-homogeneous plate (A) and reference (B)

Conclusion

The proposed method has proven effective for solving inverse problems, accurately identifying structures even without prior knowledge of boundary conditions. More specifically, the main contributions of this paper are:

- the extension of the PINN framework presented in [19] for inverse identification of a spatially rapidly-varying complex elastic modulus in an Euler-Bernoulli beam. A bounded plateaus parameterization that separates geometry from amplitude levels has been introduced. The parameter network is a shallow, single-hidden-layer sigmoid network whose units coincide with plateaus edges, while the state networks remain standard smooth multilayer perceptrons. The learnable plateaus deliver interpretable fields with plateau-like regions, localized loss, and expected levels outside the patch, all while preserving accurate displacement fits. Embedding simple geometric structure directly into the material field thus turns good fits into credible reconstructions under sparse sensing and high-order physics. Future work will investigate robustness to noise, multi-frequency excitation, experimental validation and automatic selection of the number of plateaus.
- the application of the framework introduced in [19] for the inverse analysis of thin plates. For homogeneous damped plates, the networks reliably estimate the Young's modulus and the loss factor. While exact predictions for non-

homogeneous plates remain challenging, the algorithm successfully identified damaged regions, highlighting its potential for structural health monitoring. Future work will focus on fully experimental datasets, but also to the extension of the bounded plateaus approach for 2D applications so that rapidly-varying local changes in the mechanical properties can be quantitatively computed by the machine learning tool.

Acknowledgments The authors acknowledge the support of EIPHI (ANR-17-EURE-0002).

References

1. Cross, E.J., Gibson, S.J., Jones, M.R., Pitchforth, D.J., Zhang, S., and Rogers, T.J. *Physics-Informed Machine Learning for Structural Health Monitoring*, pages 347–367. Springer International Publishing, Cham (2022)ifnextchar.gobble.
2. Aguirre, L.A. *An Introduction to Nonlinear System Identification*, pages 133–154. Springer Nature Switzerland, Cham (2024)ifnextchar.gobble.
3. Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. "Physics-informed machine learning". *Nature Reviews Physics*, 3(6):422–440 (2021)ifnextchar.gobble.
4. Raissi, M., Perdikaris, P., and Karniadakis, G. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". *Journal of Computational Physics*, 378:686–707 (2019)ifnextchar.gobble.
5. Henkes, A., Wessels, H., and Mahnken, R. "Physics informed neural networks for continuum micromechanics". *Computer Methods in Applied Mechanics and Engineering*, 393:114790 (2022)ifnextchar.gobble.
6. Amini Niaki, S., Haghighat, E., Campbell, T., Poursartip, A., and Vaziri, R. "Physics-informed neural network for modelling the thermochemical curing process of composite-tool systems during manufacture". *Computer Methods in Applied Mechanics and Engineering*, 384:113959 (2021)ifnextchar.gobble.
7. Mao, Z., Jagtap, A.D., and Karniadakis, G.E. "Physics-informed neural networks for high-speed flows". *Computer Methods in Applied Mechanics and Engineering*, 360:112789 (2020)ifnextchar.gobble.
8. Rezaei, S., Harandi, A., Moeineddin, A., Xu, B.X., and Reese, S. "A mixed formulation for physics-informed neural networks as a potential solver for engineering problems in heterogeneous domains: Comparison with finite element method". *Computer Methods in Applied Mechanics and Engineering*, 401:115616 (2022)ifnextchar.gobble.
9. As'ad, F. and Farhat, C. "A mechanics-informed deep learning framework for data-driven nonlinear viscoelasticity". *Computer Methods in Applied Mechanics and Engineering*, 417:116463 (2023)ifnextchar.gobble.
10. Zhai, W., Tao, D., and Bao, Y. "Parameter estimation and modeling of nonlinear dynamical systems based on Runge–Kutta physics-informed neural network". *Nonlinear Dynamics*, 111(22):21117–21130 (2023)ifnextchar.gobble.
11. Lai, Z., Mylonas, C., Nagarajaiah, S., and Chatzi, E. "Structural identification with physics-informed neural ordinary differential equations". *Journal of Sound and Vibration*, 508:116196 (2021)ifnextchar.gobble.
12. Di Lorenzo, D., Champaney, V., Marzin, J., Farhat, C., and Chinesta, F. "Physics informed and data-based augmented learning in structural health diagnosis". *Computer Methods in Applied Mechanics and Engineering*, 414:116186 (2023)ifnextchar.gobble.
13. Zhou, W. and Xu, Y. "Damage identification for plate structures using physics-informed neural networks". *Mechanical Systems and Signal Processing*, 209:111111 (2024)ifnextchar.gobble.
14. Kapoor, T., Wang, H., Núñez, A., and Dollevoet, R. "Physics-informed neural networks for solving forward and inverse problems in complex beam systems". *IEEE Transactions on Neural Networks and Learning Systems*, page 1–15 (2023)ifnextchar.gobble.
15. Faroughi, S., Darvishi, A., and Rezaei, S. "On the order of derivation in the training of physics-informed neural networks: case studies for non-uniform beam structures". *Acta Mechanica*, 234(11):5673–5695 (2023)ifnextchar.gobble.
16. Wu, W., Daneker, M., Jolley, M.A., Turner, K.T., and Lu, L. "Effective data sampling strategies and boundary condition constraints of physics-informed neural networks for identifying material properties in solid mechanics". *Applied Mathematics and Mechanics*, 44(7):1039–1068 (2023)ifnextchar.gobble.
17. Pezerat, C. and Guyader, J. "Force analysis technique: reconstruction of force distribution on plates". *Acta Acustica united with Acustica*, 86(2):322–332 (2000)ifnextchar.gobble.
18. Leclère, Q., Ablitzer, F., and Pézerat, C. "Practical implementation of the corrected force analysis technique to identify the structural parameter and load distributions". *Journal of Sound and Vibration*, 351:106–118 (2015)ifnextchar.gobble.
19. Teloli, R.d.O., Tittarelli, R., Bigot, M., Coelho, L., Ramasso, E., Le Moal, P., and Ouisse, M. "A physics-informed neural networks framework for model parameter identification of beam-like structures". *Mechanical Systems and Signal Processing*, 224:112189 (2025)ifnextchar.gobble.
20. Ungar, E.E., Ross, D., and Kerwin Jr, E.M. "Damping of flexural vibrations by alternate viscoelastic and elastic layers". Technical report (1959)ifnextchar.gobble.
21. Timoshenko, S. and Woinowsky-Krieger, S. "Theory of plates and shells" (1959)ifnextchar.gobble.

Chapter 3
Neural Operators for Accelerating Flow Field Predictions in an Empty Duct

Alan Xavier and Ludovic Renson

Abstract Scientific machine learning leverages modern machine learning (ML) methods to accelerate the analysis of dynamical systems governed by ordinary and partial differential equations. In fluid mechanics, these techniques offer the potential to approximate complex, nonlinear flow behaviour at a fraction of the cost of conventional computational fluid dynamics (CFD). While most prior approaches have relied on neural networks trained to learn mappings between finite-dimensional vector spaces, fluid flows are inherently continuous and infinite-dimensional. Neural operators (NOs) address this challenge by learning mappings directly between function spaces, enabling generalization across a wide range of boundary conditions and operating regimes.

In this work, we develop a neural operator framework based on the Deep Operator Network (DeepONet) to predict unsteady flow fields in a simplified aeroacoustic configuration: an empty duct with an outlet pressure boundary condition that includes a sinusoidal fluctuation. This fluctuating outlet pressure represents upstream-propagating acoustic waves, a canonical test problem for aeroelastic and turbomachinery flow analyses. High-fidelity CFD simulations are performed across a range of forcing frequencies. From these data, our DeepONet model — augmented with recurrent layers to capture temporal evolution — learns to map the transient response of the first oscillation cycle to the steady response of the final cycle.

This approach bypasses the need to simulate every cycle of the unsteady flow, reducing computational costs by an order of magnitude while retaining accuracy in the predicted density, pressure and velocity fields. The resulting operator can provide rapid flow field predictions, demonstrating a pathway for embedding scientific machine learning into turbomachinery design workflows where repeated evaluations of unsteady flow solutions are required.

Keywords Machine Learning · Neural Operators · Deep Learning · Aerodynamics · Turbomachines

Introduction

Computational fluid dynamics (CFD) is indispensable in the design and analysis of turbomachinery, where accurate predictions of unsteady flows, blade–row interactions, and aeroelastic phenomena are essential for both performance and safety. However, the fidelity required to capture the strongly coupled flow physics in rotating machinery comes at a significant computational cost. High-resolution unsteady simulations, such as unsteady Reynolds-averaged Navier–Stokes (URANS) or large eddy simulations (LES), may require thousands of CPU hours for a single operating condition. As a result, comprehensive parametric sweeps, real-time aeroelastic stability assessments, and large-scale uncertainty quantification remain infeasible within typical industrial design cycles.

Neural operators (NOs) offer a promising route to overcome these limitations by directly learning the solution operators of the governing partial differential equations. Architectures such as Fourier Neural Operators (FNOs) [1] and Deep Operator Networks (DeepONets) [2] can approximate flow fields across varying boundary conditions, blade geometries, and operating regimes. Once trained, these models can deliver high-fidelity flow predictions at a fraction of the cost of conventional solvers, enabling rapid evaluation of unsteady aerodynamic loads and modal responses. In turbomachinery applications, this

Alan Xavier · Ludovic Renson
Department of Mechanical Engineering
Imperial College London, South Kensington Campus, London SW7 2AZ, UK
e-mail: aax23@ic.ac.uk; l.renson@imperial.ac.uk

© The Author(s), under exclusive license to River Publishers 2026

Kristin B. Zimmerman, PhD (eds.), *Data Science in Engineering, 2026, Vol. 9*, Conference Proceedings
of the Society for Experimental Mechanics Series, https://doi.org/10.13052/rp-9788743814207

capability opens the door to accelerating flutter analysis, forced-response prediction, and multi-point performance optimization.

A key advantage of NOs is their ability to generalize across parametric variations without requiring retraining for each new case. This property makes them particularly well-suited for design space exploration and probabilistic risk assessment during early-stage engine development. By embedding NO-based surrogates into CFD and aeroelastic workflows, the industry can achieve both dramatic computational acceleration and enhanced predictive capability—bridging the gap between the demands of high-fidelity simulation and the real-time decision-making needs of modern propulsion system design.

Methods

We consider the prediction of unsteady flow fields in an empty duct of length L and constant cross-section, governed by the unsteady compressible Navier–Stokes equations. Different operating conditions are simulated by varying the outlet boundary condition, where a sinusoidally varying pressure wave of amplitude 300 kPa and frequency ranging from 800 Hz to 1200 Hz (in increments of 25 Hz) is applied. The inlet is modeled at atmospheric conditions to allow upstream propagation of the pressure wave. Inviscid wall conditions are enforced on the top and bottom surfaces. This canonical setup emulates aeroacoustic and aeroelastic test problems, serving as a baseline for validating operator-learning methods in unsteady turbomachinery flows.

Let $\mathbf{u}(x, y, z, t)$ denote the flow state, consisting of velocities (U_x, U_y, U_z), pressure p, and density ρ, at spatial coordinates (x, y, z) and time t. Our objective is to approximate the solution operator

$$\mathcal{G} : \mathcal{I} \mapsto \mathbf{u}(x, y, z, t),$$

which maps the transient flow field over the first oscillation period to the flow field of the final period. Instead of advancing the governing PDEs numerically, we approximate $\mathcal{G}$ using a Deep Operator Network (DeepONet) with recurrent dynamics in the branch net to capture temporal correlations.

The DeepONet framework is decomposed into two subnetworks:

1. **Branch Network (LSTM-based):** Inspired by He et al. [3, 4], the branch network encodes the temporal evolution of the inflow conditions sampled over discrete time steps during the first oscillation period. A stack of Long Short-Term Memory (LSTM) layers captures both periodicity and nonlinear dependencies of the propagating wave. This produces a latent embedding b of dimension $H \times T$, where H is the hidden dimension and T is the number of timesteps per oscillation.

2. **Trunk Network (Fully Connected):** The trunk network receives spatial coordinates (x, y, z) as input and generates a latent embedding t of dimension $N \times H \times C$, where N is the number of mesh nodes and C is the number of flow states.

3. **Operator Construction:** The operator $\mathcal{G}$ is formed by bilinear interaction of the branch and trunk embeddings:

$$\mathcal{G} = \sum_{h=1}^{H} b_{h,t}\, t_{n,h,c},$$

where $n \in [1, N]$, $h \in [1, H]$, and $c \in [1, C]$. This formulation ensures that the LSTM branch encodes inlet-driven temporal dynamics, while the trunk network captures spatial response modes of the duct.

Training Strategy

High-fidelity CFD data from the first oscillation period ($0 \leq t \leq T = 1/f$) is used for supervised training. The duct response over this window contains both the primary wave propagation and any nonlinear distortions. Since the flow is uniform in the z-direction, only samples in the xy-plane are considered for training.

To reduce input dimensionality, we sample node locations along the top wall for the branch network, as no fluctuations are observed in the y-direction. Furthermore, instead of using all five primitive variables as inputs, we compute the Mach number M, defined as

$$M = \frac{V}{a}, \quad a = \sqrt{\gamma \frac{p}{\rho}}, \quad V = \sqrt{U_x^2 + U_y^2 + U_z^2},$$

which compactly encodes the flow state. Inlet and outlet variables are normalized to $[-1, 1]$ using min–max scaling to eliminate scale discrepancies between features.

The model is trained to minimize the mean squared error (MSE) between predicted and reference CFD solutions across spatial and temporal dimensions for each flow variable:

$$\mathcal{L} = \sum_{c=1}^{C} \mathcal{L}_c, \quad c \in \{\rho, U_x, U_y, U_z, p\},$$

where each component loss is defined as

$$\mathcal{L}_c = \frac{1}{N_x N_y N_z N_t} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \sum_{k=1}^{N_z} \sum_{l=1}^{N_t} \left\| u(x_i, y_j, z_k, t_l) - \hat{u}(x_i, y_j, z_k, t_l) \right\|_2^2.$$

After training, the DeepONet–LSTM model is deployed to predict the flow field over the final oscillation period ($(S-1)T < t \leq ST$), for S total periods.

Results

The model was trained using mini-batches of forcing sequences covering multiple operating conditions. Optimization was performed with the AdamW optimizer [5], employing a learning rate of 1×10^{-4} and weight decay regularization. Training was conducted for 5000 epochs.

We evaluated two model variants:

1. **CFD mesh model** — trained directly on the unstructured CFD solver mesh.
2. **Structured mesh model** — trained on a uniform grid of 200×20 points in the axial and vertical directions, respectively. This structured mesh contains approximately four times fewer nodes than the unstructured CFD mesh, reducing computational expense.

Figure 1 compares the DeepONet–LSTM predictions with the reference CFD solutions for the final timestep of the last oscillation period in the duct, across three flow variables: density (ρ), axial velocity (U_x), and pressure (p).

- **Overall prediction quality:** The predicted fields (top row) closely reproduce the structure and periodicity of the CFD reference solutions (middle row). The spatial oscillations induced by the sinusoidal outlet pressure condition are captured with high fidelity across the duct length.
- **Density field (ρ):** The DeepONet model reconstructs the wave pattern with strong agreement in both wavelength and amplitude. The difference field (bottom row) shows only small discrepancies, localized near the outlet boundary (left) where the gradients are steepest.

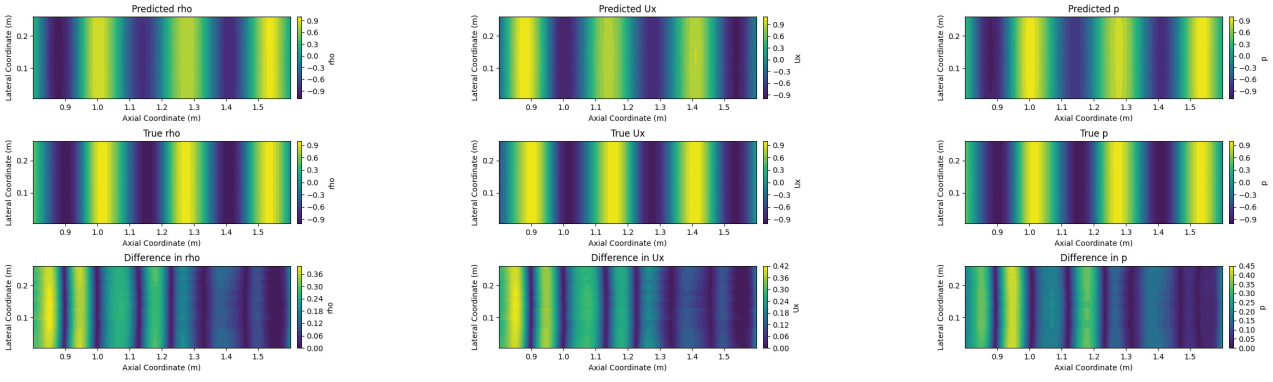


Fig. 1 Predicted flow fields for the final timestep of the last oscillation period on the **unstructured CFD mesh**. Results are shown for (a) density ρ, (b) axial velocity U_x, and (c) pressure p. For each variable, the top row shows the DeepONet–LSTM prediction, the middle row shows the reference CFD solution, and the bottom row shows the absolute difference field.

- **Axial velocity (U_x):** Predictions align well with the CFD solution, especially near the inlet (right). Minor deviations appear closer to the outlet (left), suggesting sensitivity of the network to boundary reflections or phase shifts in the velocity response.
- **Pressure field (p):** The predicted pressure oscillations match the reference solution in both phase and amplitude. Errors are again localized near the downstream boundary (left), but their magnitude remains small relative to the overall oscillation amplitude.
- **Error characteristics:** Across all three fields, the difference plots confirm that the DeepONet–LSTM model captures the dominant wave dynamics while underpredicting fine-scale variations near the outlet (left). The overall error magnitudes (≤ 0.45 in normalized units) remain modest, showing that the model generalizes well across the spatial domain.

In short, the DeepONet–LSTM framework accurately predicts the periodic wave propagation in the empty duct test case. It reproduces the dominant spatial oscillations in density, velocity, and pressure, with only minor boundary-localized discrepancies.

Figure 2 presents the DeepONet–LSTM predictions against the reference CFD solutions for the final timestep of the last oscillation period in the duct, this time on a **structured mesh (200 × 20 grid)**. As in Figure 1, results are shown for density (ρ), axial velocity (U_x), and pressure (p).

- **Overall prediction quality:** The model predictions (top row) reproduce the CFD fields (middle row) with high accuracy, and the error magnitudes (bottom row) are noticeably lower than those obtained with the unstructured mesh in Figure 1. The structured grid appears to aid the network in learning smoother and more consistent spatial patterns.
- **Density field (ρ):** The predicted density field closely follows the CFD solution across the duct. The difference field shows very small discrepancies (< 0.3 in normalized units), primarily confined to the downstream boundary (left). Compared to the unstructured mesh case, the error amplitude is reduced by roughly 25%.
- **Axial velocity (U_x):** Predictions match the CFD reference almost exactly along the duct axis. The absolute error field is smooth and minimal, indicating that the structured mesh improved the model's ability to capture phase alignment and amplitude of the oscillatory velocity field.
- **Pressure field (p):** The sinusoidal pressure oscillations are very well captured in both wavelength and amplitude. The difference field remains small and uniform across most of the duct, with maximum discrepancies near the outlet (left) but still significantly reduced compared to Figure 1.
- **Error characteristics:** The errors are uniformly small across the domain, with the largest magnitudes reduced by roughly half of those in the unstructured mesh case. The structured discretization provides cleaner spatial inputs, which likely allows the DeepONet–LSTM model to better capture the propagating wave dynamics.

In summary, the structured mesh results (Figure 2) show a clear improvement over the unstructured mesh case (Figure 1). The DeepONet–LSTM model not only reproduces the dominant flow oscillations but also achieves much smaller and more spatially uniform errors. This confirms that training on structured grids offers a more efficient and accurate surrogate for unsteady wave propagation in ducts.

We observe that the structured mesh model achieves improved accuracy compared with the unstructured mesh model, while also offering slightly reduced training times (24.6 hours as compared to 26.5 hours respectively). This tradeoff suggests that structured discretizations may be advantageous for operator-learning approaches, particularly when scaling to more complex turbomachinery blade-row flows. However, further investigation is required to confirm whether similar gains persist in regions of rapid spatial variation, such as blade wakes or shock–boundary layer interactions, where fine resolution is typically necessary in traditional CFD.

A comparison of Figures 1 and 2 highlights the tradeoffs between the two training strategies. The unstructured mesh model (Figure 1) reproduces the main wave features but exhibits localized discrepancies near the duct outlet (left), where wave steepening and reflections occur. By contrast, the structured mesh model (Figure 2) shows reduced error magnitudes across the duct, particularly in the downstream region, suggesting that the regular discretization aids the network in learning coherent spatial modes. However, both models capture the dominant oscillatory dynamics with comparable fidelity, indicating that the structured mesh provides a more computationally efficient representation without sacrificing predictive accuracy. This efficiency is especially promising for scaling to turbomachinery blade-row problems, where the cost of training on full CFD meshes is prohibitive.

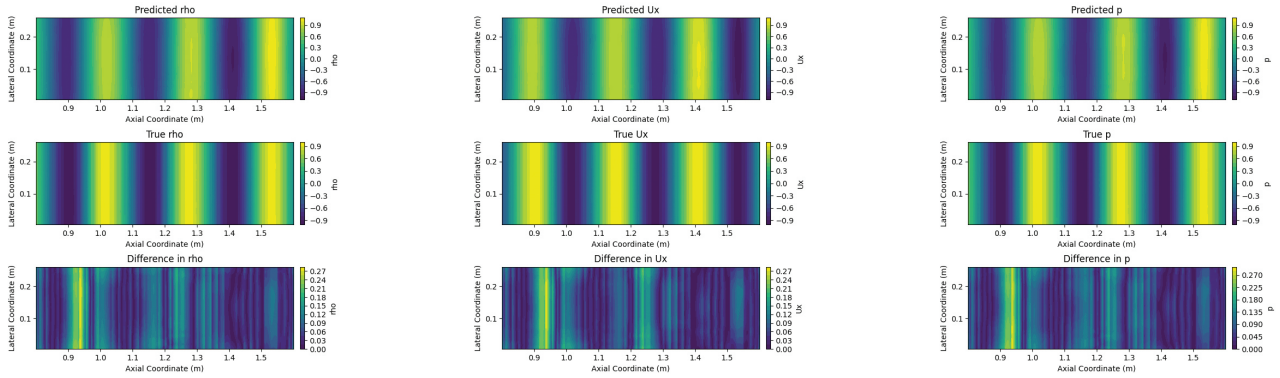


Fig. 2 Predicted flow fields for the final timestep of the last oscillation period on the **structured mesh**(200×20 grid). Results are shown for (a) density ρ, (b) axial velocity U_x, and (c) pressure p. The top row shows the DeepONet–LSTM prediction, the middle row shows the CFD reference solution, and the bottom row shows the absolute difference field.

Finally, we computed normalized error measures across the duct domain for each flow variable, across the final oscillation period, and model variant:

1. **Normalized Root-Mean-Square Error (NRMSE):**

$$\mathrm{NRMSE}_c = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(u_c(x_i) - \hat{u}_c(x_i)\right)^2}}{u_{c,\max} - u_{c,\min}}$$

   where u_c and $\hat{u}_c$ denote the CFD and DeepONet–LSTM predictions for flow variable c, and the denominator normalizes by the dynamic range of the variable.

2. **Relative L_2 Error:**

$$E_{L_2,c} = \frac{\|u_c - \hat{u}_c\|_2}{\|u_c\|_2}$$

Table 1 quantifies the predictive accuracy of the DeepONet–LSTM models trained on the unstructured CFD mesh and the structured mesh. On average, the structured mesh reduces the NRMSE from **9.52% to 4.46%** and the relative L_2 error from **0.277 to 0.129**. This represents more than a twofold improvement in accuracy across all flow variables. The gains are consistent for density, velocity, and pressure, indicating that the structured discretization provides a more uniform representation of the spatial dynamics, which the network can exploit to better capture the wave propagation. These results confirm that structured meshes not only accelerate training but also enhance predictive fidelity, making them particularly attractive for extending this framework to more complex turbomachinery configurations.

Table 1 Comparison of error metrics between the DeepONet–LSTM models trained on the unstructured CFD mesh and the structured mesh (200×20 grid).

Variable	**CFD mesh**		**Structured mesh**	
	NRMSE [%]	E_{L_2}	NRMSE [%]	E_{L_2}
Density (ρ)	9.37	0.273	4.49	0.130
Axial velocity (U_x)	9.61	0.279	4.35	0.126
Pressure (p)	9.58	0.279	4.54	0.132
Average	9.52	0.277	4.46	0.129

Conclusion

We have demonstrated the application of a DeepONet architecture with an LSTM-based branch network for predicting unsteady flow fields in an empty duct subjected to sinusoidal outlet forcing. Trained on a single oscillation period, the model accurately reconstructs the flow field during the final period, effectively bypassing thousands of CFD timesteps.

Quantitative error analysis confirms the predictive capability of the approach. When trained on the unstructured CFD mesh, the DeepONet–LSTM model achieved average errors of 9.52% (NRMSE) and 0.277 (L_2). Training on a structured mesh reduced these values by more than half, to 4.46% and 0.129, respectively. The structured discretization thus not only accelerated training but also enhanced predictive fidelity, yielding smoother and more accurate reconstructions of the wave dynamics.

The DeepONet–LSTM framework therefore provides rapid, accurate predictions of unsteady pressure distributions and aerodynamic loads, enabling orders-of-magnitude acceleration relative to conventional CFD. These surrogates can be integrated directly into aeroelastic solvers for flutter margin prediction or leveraged in parametric studies requiring large numbers of operating conditions.

This empty-duct case study serves as a controlled validation of the operator's stability and generalization. Future work will extend the framework to turbomachinery blade-row configurations, where richer unsteady features such as wakes, shocks, and blade interactions must be captured. A natural progression is to benchmark the DeepONet–LSTM model against Fourier Neural Operators (FNOs) [1] on canonical cascade test cases, such as Standard Configuration 10, which supports both subsonic and transonic flow regimes.

Acknowledgments The authors would like to thank Rolls-Royce Plc. for providing the financial support for this project (iCASE #230125) and for giving permission to publish this work. The authors also acknowledge the financial support of the EPSRC (grant #EP/Y528560/1).

References

1. Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. "Fourier neural operator for parametric partial differential equations" (2021)ifnextchar.gobble.
2. Lu, L., Jin, P., Pang, G., Zhang, Z., and Karniadakis, G.E. "Learning nonlinear operators via deeponet based on the universal approximation theorem of operators". *Nature Machine Intelligence*, 3(3):218–229 (2021)ifnextchar.gobble.
3. He, J., Kushwaha, S., Park, J., Koric, S., Abueidda, D., and Jasiuk, I. "Sequential deep operator networks (s-deeponet) for predicting full-field solutions under time-dependent loads". *Engineering Applications of Artificial Intelligence*, 127:107258 (2024)ifnextchar.gobble.
4. He, J., Kushwaha, S., Park, J., Koric, S., Abueidda, D., and Jasiuk, I. "Predictions of transient vector solution fields with sequential deep operator network". *Acta Mechanica*, 235(8):5257–5272 (2024)ifnextchar.gobble.
5. Loshchilov, I. and Hutter, F. "Decoupled weight decay regularization" (2019)ifnextchar.gobble.

Chapter 4
Shaping a Shock Response Spectrum via Metamaterial Topology Optimization

Brannon Boyd and Vijaya V. N. Sriram Malladi

Abstract This paper investigates the geometric effects of a metamaterial beam on shock response spectrum (SRS) amplitudes. We use a two-dimensional Timoshenko beam model, with a unit-cell sequence repeated 20 times along its length, and simulated its behavior using a finite element method (FEM) code. The model, consisting of N nodes and 2N degrees-of-freedom (DOF), was subjected to a shock load that propagated temporally through the structure. For comparison, a uniform, constant-thickness beam was used as a control case.

The shock response spectrum was computed for the deflection DOF at each node. An optimization study was then conducted using a Gaussian Process (GP) Bayesian optimization scheme. The parameter space consisted of the duty cycle, number of periods, and individual amplitudes with the objective function to minimize the area under the SRS curve within a specified frequency band. The resulting optimized metamaterial structure demonstrated substantial attenuation of a hypothetical single degree-of-freedom (SDOF) response, not only within the targeted frequency band of 1-2 kHz, but across the entire frequency range. Analysis of the transmission spectrum confirmed that the structural bandgaps were shifted to an ideal case, enabling the effective attenuation of the rapid transient input load. This shows the potential for further SRS shaping and tuning to achieve frequency-specific responses.

Keywords Metamaterials · Shock Response Spectrum · Bayesian Optimization · Timoshenko Beam · Bandgap

Glossary

DOF degrees-of-freedom.
EI expected improvement.
FEM finite element method.
GP gaussian process.
LR local resonance.
PVSS pseudo-velocity shock spectrum.
SDOF single degree-of-freedom.
SRS shock response spectrum.

Introduction

The analysis of beams is fundamental in understanding the underlying dynamics of systems in structural mechanics, with applications ranging from civil and aerospace engineering to biomedical engineering. More recently, metamaterials, which

Brannon Boyd · Vijaya V.N. Sriram Malladi
Vibrations, Intelligent Testing, & Active Learning of Structures (ViTALS) Group, Department of Mechanical and Aerospace Engineering, Michigan Technological University, Houghton, MI 49931
e-mail: boydbn@lanl.gov; smalladi@mtu.edu

Brannon Boyd
Los Alamos National Laboratory, Verification & Validation (V-14), Testing Development and Structural Analysis (TDSA) Team, Los Alamos National Laboratory, P.O. Box 1663, Los Alamos, NM 87545
e-mail: boydbn@lanl.gov

B. Boyd is supported by LANL funding and Vijaya V. N. Sriram Malladi is supported by NSF funding for this work.

© The Author(s), under exclusive license to River Publishers 2026

Kristin B. Zimmerman, PhD (eds.), *Data Science in Engineering, 2026, Vol. 9*, Conference Proceedings
of the Society for Experimental Mechanics Series, https://doi.org/10.13052/rp-9788743814207

are engineered materials with tailored properties achieved through periodic structural patterns, have shown great promise in controlling aspects of the dynamic behavior. This paper explores the design of a metamaterial beam engineered to mitigate shock response through strategic geometric shaping. The proposed methodology integrates FEM for discretizing and simulating the system, eigenvalue analysis to identify modal characteristics, a time integration scheme for simulating transient shock responses, and Gaussian process (GP) optimization to refine the beam's geometry for optimal performance within desired frequency bands.

Background

The control of mechanical waves, specifically for applications in vibration and shock isolation, is a central challenge in structural dynamics. Typical design methodologies use passive methods, such as modifying damping or increasing the gross structural rigidity to shift the fundamental frequency. However, these methods are often limited in their efficacy, especially when dealing with broadband transient inputs, and can lead to undesirably heavy or rigid structures and limit the ability to use joints.

Mechanical metamaterials have enabled new ways of understanding structural dynamics. Metamaterials are a class of engineered materials whose properties are derived from their intricate geometric design rather than their base material composition [5, 6]. A key feature of these materials are periodic structures, which enable the formation of phononic bandgaps—theoretical frequency ranges where elastic waves cannot propagate. This phenomenon, which can arise from Bragg scattering (due to periodicity) or local resonance (LR) (due to embedded oscillators), has been extensively studied for its potential to create highly effective vibration isolation systems [1, 2, 9]. A significant body of work has focused on designing these structures to create or shift bandgaps to targeted frequencies, with many studies demonstrating their effectiveness in attenuating steady-state vibrations [9].

More recently, the application of metamaterials has expanded to shock attenuation, which involves a rapid, broadband transient input. The SRS has become the standard metric for quantifying a structure's resilience to such loads [3, 4]. The metric involves the evaluation of a peak response, when modeled as a SDOF oscillator. This paper does not seek to evaluate the validity of the metric, but only use it for comparative purposes. Although several studies have proposed metamaterial designs for shock mitigation, most have focused on materials with negative stiffness or geometric nonlinearities to dissipate energy. Few studies have explored the purely geometric effects of periodic structures on broadband shock response, particularly the direct link between geometric parameters or added LRs, bandgap formation, and the resulting SRS. This gap in the literature represents a critical area for research, as it can lead to reusable, printable, all-elastic designs that do not rely on material dissipation mechanisms.

Furthermore, the design of these geometrically complex structures for a specific performance objective, such as a reduction in SRS amplitudes, constitutes the need for a high-dimensional optimization problem. Traditional gradient-based optimization methods are often ill-suited for this task due to the high computational cost of non-trivial geometries in finite element simulations and the often non-smooth nature of the objective function. Consequently, there is an interest in using more advanced optimization techniques that predict the exact solution with well-defined uncertainty. Bayesian optimization, which uses a GP to build a surrogate model of the objective function, is often employed due to its sample efficiency, making it ideal for problems with expensive black-box function evaluations [7, 10]. While Bayesian optimization has been successfully applied to other metamaterial design problems [8], its application to the specific challenge of minimizing an SRS is a novel contribution.

This paper fits into the existing literature by addressing the underexplored realm of geometric metamaterial design, shock response attenuation, and advanced optimization. We combine a first-order Timoshenko beam finite element methods model with a volume-constrained geometric parameterization and a GP-based Bayesian optimization scheme. This work demonstrates the potential feasibility of tailoring metamaterial beams for superior shock performance and also provides a systematic, computationally efficient methodology for their design.

Methods

The Control Beam

Most mechanical systems with non-trivial displacement are governed by the standard second-order linear dynamic equation of motion:

$$\mathbf{M}\ddot{x}(t) + \mathbf{C}\dot{x}(t) + \mathbf{K}x(t) = \mathrm{F}(t) \tag{1}$$

where $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{K}$ are the global mass, damping, and stiffness matrices, respectively. The vector $x(t)$ represents the nodal displacements (along with its derivatives $\dot{x}(t)$ and $\ddot{x}(t)$), and $\mathrm{F}(t)$ is the external force vector due to applied temporal load.

To quantify the dynamic performance improvements of the optimized metamaterial beam, a fundamental control case was established. This baseline model consisted of a uniform, constant-thickness first-order Timoshenko beam with a simple, homogeneous cross-section. Timoshenko beam theory is typically used for shorter beams as it accounts for both shear deformation and rotational inertia. This is a more accurate model for short, thick beams or for higher-frequency modes, where the Euler-Bernoulli assumption (plane sections remain plane and normal to the deformed axis) breaks down. However, the beam being analyzed herein is long and slender, but because of the periodic nature of a metamaterial, the wave propagation through the unit-cell and dispersion relation requires a Timoshenko beam to effectively model the physics.

The elemental mass matrix, $\mathbf{M}_e$, is a combination of the translational and rotational inertia:

$$\mathbf{M}_e = \mathbf{M}_{11} + \mathbf{M}_{22}$$

where

$$\mathbf{M}_{11} = \frac{\rho A h_e}{6}\begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{M}_{22} = \frac{\rho I h_e}{6}\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2 \end{bmatrix}.$$

Here, ρ is the density, A is the cross-sectional area, I is the second moment of area, and h_e is the element length. The first matrix, $\mathbf{M}_{11}$, represents the translational inertia, while the second, $\mathbf{M}_{22}$, accounts for the rotational inertia.

The elemental stiffness matrix, $\mathbf{K}_e$, includes contributions from both bending and shear deformation:

$$\mathbf{K}_e = \mathbf{K}_{11} + \mathbf{K}_{12} + \mathbf{K}_{21} + \mathbf{K}_{22}$$

where

$$\mathbf{K}_{11} = \frac{K_f G A}{h_e}\begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \qquad \mathbf{K}_{12} = -\frac{K_f G A}{2}\begin{bmatrix} 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{K}_{21} = -\frac{K_f G A}{2}\begin{bmatrix} 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \end{bmatrix}; \quad \mathbf{K}_{22} = \frac{EI}{h_e}\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} + \frac{K_f G A h_e}{4}\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}.$$

Here, E is Young's modulus, G is the shear modulus, and K_f is the shear correction factor, $K_f = \left(\frac{0.87+1.12\,\nu}{1+\nu}\right)^2$. The matrices $\mathbf{K}_{11}$, $\mathbf{K}_{12}$, and $\mathbf{K}_{21}$ are related to the shear deformation, while $\mathbf{K}_{22}$ accounts for both the bending stiffness (EI/h_e) and an additional stiffness contribution from the shear deformation. The sum of these sub-matrices forms the full elemental stiffness matrix. The design of this control beam was crucial for ensuring a valid comparative analysis, isolating the effects of the metamaterial's geometric features from other variables. Rayleigh damping was used and modeled using the first two natural frequencies ω_1 and ω_2:

$$\alpha = \frac{2\zeta\omega_1\omega_2}{\omega_1+\omega_2}, \qquad \beta = \frac{2\zeta}{\omega_1+\omega_2}, \qquad C = \alpha\tilde{M} + \beta\tilde{K}, \tag{2}$$

where ζ is the target damping ratio and C is the damping matrix.

The control beam was modeled such that the metamaterial beam would have identical material properties, specifically the same Young's modulus (E), Poisson's ratio (ν), and density (ρ), as the control beam. Its total length, mass, and width were also kept consistent with the metamaterial counterpart. However, the most critical difference was the control beam maintained a constant thickness amplitude across the length. By maintaining this mass equivalence, any observed differences in the SRS amplitudes could be attributed solely to the geometric configuration of the beam, rather than to differences in total mass or nodal DOFs.

The control beam was subjected to the same temporal shock load as the metamaterial beam. A classical half-sine pulse was used for simplicity with an amplitude of 100 Gs and duration of 2.1 ms. The time waveform was propagated through the structure using a Newmark-beta integration technique due to its unconditional stability. Generally, the method updates the state vectors at each time step $i+1$ based on the values at the previous time step i, using the following relationships:

$$\dot{\mathbf{x}}_{i+1} = \dot{\mathbf{x}}_i + \Delta t\left[(1-\gamma)\ddot{\mathbf{x}}_i + \gamma\ddot{\mathbf{x}}_{i+1}\right]$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \Delta t \dot{\mathbf{x}}_i + \Delta t^2 \left[\left(\frac{1}{2} - \beta \right) \ddot{\mathbf{x}}_i + \beta \ddot{\mathbf{x}}_{i+1} \right]$$

where Δt is the time step, and β and γ are Newmark-Beta parameters. Here we used the constant average acceleration method with $\beta = 0.25$ and $\gamma = 0.5$ due to not needing high-frequency algorithmic damping and desired unconditionally stability.

Finally, the shock response calculation was done with a Tustin bilinear filter using Smallwood's ramp invariant method. The selected objective metric for shock was the SRS. It is a plot of the maximum acceleration response of a set of SDOF systems to a transient input, as a function of their natural frequency. Each SDOF system represents a specific natural frequency and damping ratio. The SRS is a standard tool for quantifying the severity of a shock event on a structure. The transfer function for an SDOF system is given by:

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

where ω_n here is the $n-th$ hypothetical natural frequency of the system when modeled as an SDOF oscillator subject to base excitation.

Design of 1D Metamaterial Beam

The metamaterial beam used all the same general methods as the control beam, with the exception that the beam's topology was periodically altered. From the Timoshenko beam equations, it can be seen that changing the topology of the beam *will* alter the mass, $\mathbf{M_e}$, and the stiffness $\mathbf{K_e}$ through the cross-sectional area, A, and the area moment of inertia, I. Therefore for a more one-to-one comparison the mass must be equated which can be accomplished by holding density fixed and conserving the volume. Thus to ensure mass (volume) conservation when applying topological optimization, we normalize the raw thickness array so that the average thickness equals a reference base thickness t_{base}. The unnormalized thickness values, t_i, where $i = 1, 2, \ldots, N$, and N is the total number of nodes, can be represented as:

$$\bar{t}_{\text{raw}} = \frac{1}{N} \sum_{i=1}^{N} t_i$$

To preserve total volume (assuming unit width), we scale the entire profile by the normalization factor $t_{\text{base}}/\bar{t}_{\text{raw}}$. Thus, the normalized thickness profile, t_i^{norm} is:

$$t_i^{\text{norm}} = t_i \cdot \left(\frac{t_{\text{base}}}{\bar{t}_{\text{raw}}} \right)$$

ensuring that $\frac{1}{N} \sum_{i=1}^{N} t_i^{\text{norm}} = t_{\text{base}}$ which means the beam with spatially varying thickness has the same average thickness (and hence the same total volume) as a beam of uniform thickness t_{base}. With this simple scaling, the only other adjustment that need be made is explicitly forming the cross-sectional area and area moment of inertia ($\frac{1}{12} bh^3$) with the thickness area. Representative geometry of the control beam and an example of a metamaterial beam can be found in Fig. 1.

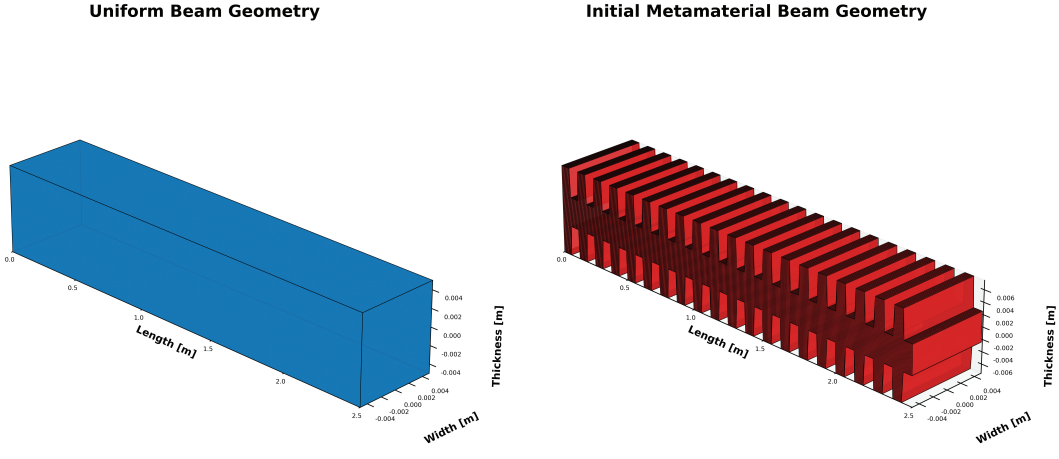


Fig. 1 Shown on the left is a 2.5 meter beam of uniform thickness, constant along the length. The beam depicted on the right is an initial metamaterial beam before geometrically optimizing via a Bayesian optimization technique. Both beams use the same material properties and have identical volume and are therefore mass-equivalent.

Optimization

We set out to optimize the initial metamaterial beam geometry using a Bayesian optimization approach. The parameter space consisted of the first section of a square wave when viewed from the side, along with the duty cycle and number of periods. The other amplitude for a full period was left unconstrained since it would need to be adjusted to conserve volume. The cost function was formed as a metric on the goodness of the metamaterial geometry under test in a simulated classical shock. Minimizing the cost function is the basis of a traditional optimization problem with the objective of reducing the beam's shock response within a specific frequency band.

With the system matrices $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{K}$ fully defined for the next y (posterior) in the objective, a numerical simulation can be performed to determine the beam's response to the specified shock loading. By numerically integrating the equations of motion for the beam's DOFs, the full SRS for all nodes can be formed. Here a modified formulation of the SRS was used, the pseudo-velocity shock spectrum (PVSS), defined as: $\text{PVSS}(f) = \frac{\text{g}\cdot\text{SRS}(f)}{2\pi f}$ where g is the gravitational constant and f is the frequency in Hertz.

The cost function is the area under the PVSS curve within the specified frequency band, from a starting frequency $f_{start} = 1000\,\text{Hz}$ to an ending frequency $f_{end} = 2000\,\text{Hz}$. The objective function can thus be expressed as:

$$\text{Objective Value} = \int_{f_{start}}^{f_{end}} \text{PVSS}(f)df \tag{3}$$

This integral must be numerically solved; here, the trapezoidal method is used. By minimizing the objective value, the optimization process reduces the peak responses of the hypothetical SDOF oscillator responses of the beam across the specified frequency range. Using a standard GP with a Bayesian optimization scheme, future points can be selected from an expected improvement (EI) acquisition function. With a GP fitted to the priors, the acquisition function determines the next optimal point to evaluate (posterior). The EI function balances the trade-off between exploring unknown regions of the design space (high variance) and exploiting areas near the current best-known minima (low mean). The EI acquisition function, which computes the expected improvement over the current best observation value y_{best}, is defined as:

$$\text{EI}(x) = E\left[\max(0,\, y_{best} - Y(x))\right] \tag{4}$$

where $Y(x)$ is a random variable representing the objective value at x, with distribution given by the GP surrogate model. The next point to evaluate, x_{next}, is found by maximizing the acquisition function:

$$x_{next} = \arg\max x \in \mathcal{X}\ \text{EI}(x) \tag{5}$$

The Bayesian process requires an adequately sized distribution of priors in order to inform the next solution within the parameter space. The initial sampling can be accomplished using a Sobol statistics method of sampling. The Sobol sampling technique is a method used to generate low-discrepancy sequences, which are particularly useful in high-dimensional spaces for numerical integration and optimization problems. Sobol sequences are designed to fill a multi-dimensional space more uniformly than random sampling techniques such as a Latin Hypercube.

The Sobol sequence is generated using a recursive formula based on binary representations of integers. For a given dimension d, the n-th Sobol point s_n can be expressed as:

$$s_n = (s_{n,1}, s_{n,2}, \ldots, s_{n,d}) \tag{6}$$

where each component $s_{n,j}$ is computed using the following formula:

$$s_{n,j} = \sum_{i=0}^{\infty} \frac{b_{i,j}}{2^{i+1}} \cdot \text{mod}(n, 2^i) \tag{7}$$

where $b_{i,j}$ are the elements of a direction number matrix that defines the Sobol sequence for dimension j. A flowchart of the general process is shown in Fig. 2.

Discussion

The results of this study show the potential for SRS manipulation via geometric metamaterial design of a beam structure. The final optimized geometry can be seen in Fig. 3, where the optimized condition was a single period — always outperforming multiple periods slightly in the cost function for this specific formulation. A reduction in the SRS amplitude for

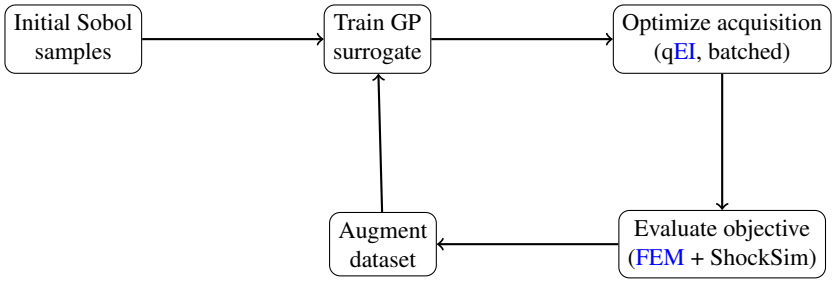


Fig. 2 Illustration of the closed-loop Bayesian optimization workflow used for design exploration: initial Sobol sampling to form the basis of priors, Gaussian process (GP) surrogate training, batched acquisition optimization using q-Expected Improvement (qEI), high-fidelity objective evaluation (FEM + ShockSim) of selected candidates, and dataset augmentation for the next optimization iteration (updated posterior from priors).

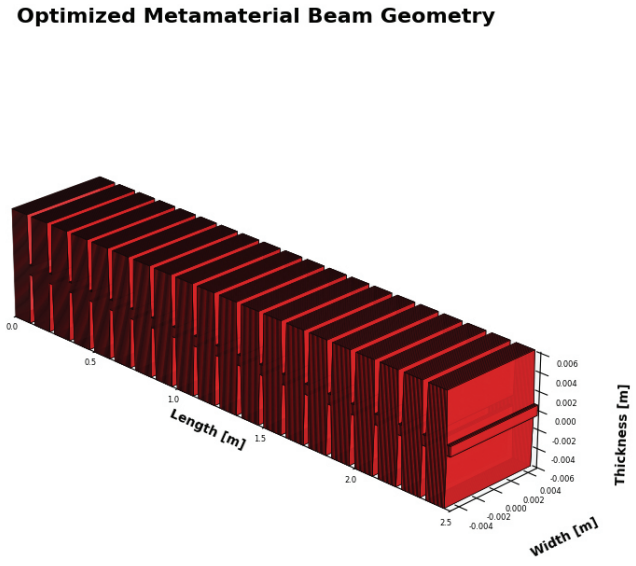


Fig. 3 Final optimized beam geometry with one period and an adjusted duty cycle and slight amplitude scaling which can be seen when comparing to Fig. 1. However, due to the single period, the optimized metamaterial is more similar to the initial metamaterial than it is different.

the optimized metamaterial beam, demonstrates the hypothesis that geometric patterning can be used to control transient mechanical waves. An exemplar case at 2.25 m response location along the length of the beam can be seen in Fig. 4 plotted on a tripartite pseudo-velocity plot. This improvement was not only to the targeted frequency band (1-2 kHz), but visible across the entire computed frequency range. The full spectral amplitude reduction suggests that the optimization process may yield an effective, broadband attenuation mechanism.

The initial metamaterial beam before optimization, as compared to the uniform beam transmission plot, is seen in Fig. 5. The transmission post optimization is also shown on the right, where it can be seen that the optimization function served to carve out the bandgap slightly more around the 1-2 kHz frequency range and shifted some of the dissipation away from the ~ 4750 Hz frequency bandgap.

Understanding the mechanisms behind these mechanics lies in the analysis of the transmission spectrum. The optimization process, driven by the objective to minimize the area under the PVSS curve, modified the beam's geometric properties to shift and broaden the structural bandgaps in the desired frequency band. The transmission spectrum of the optimized beam, when compared to the uniform control beam, showed a clear and pronounced attenuation region corresponding to the targeted frequency band. This confirms that the observed shock attenuation is a direct consequence of the wave filtering properties of the periodic structure. While the fundamental frequency (and only frequency) of the shock pulse was at 2.99 kHz, this does not mean that the impulse energy will transmit through the structure at that energy, but may have other dominant frequencies as observed here (~ 1.5 kHz). A full SRS (traditional SRS) swept along the length of the beam is depicted

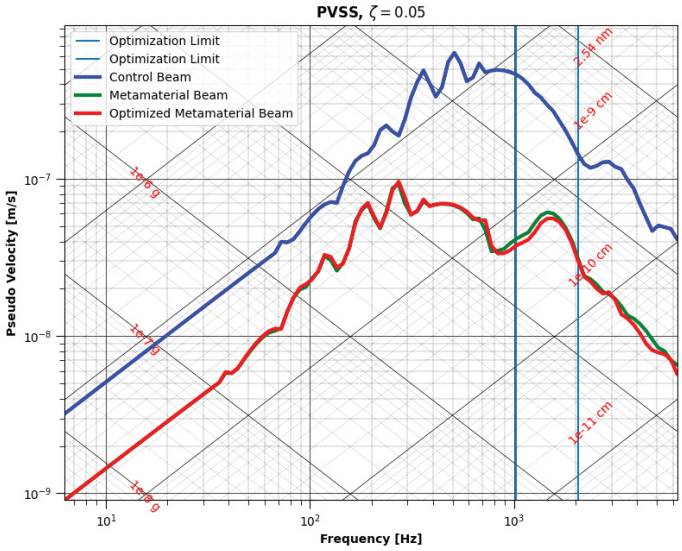


Fig. 4 Exemplar pseudo-velocity shock response spectrum for the control and different metamaterial beams at a response location of 2.25 m. The optimized metamaterial shows a slight reduction in the target frequency band at this specific location along the length of the beam. The initial metamaterial before optimization was quite close to the optimized structure — likely in part due to the simplicity of the structure and classical shock impulse.

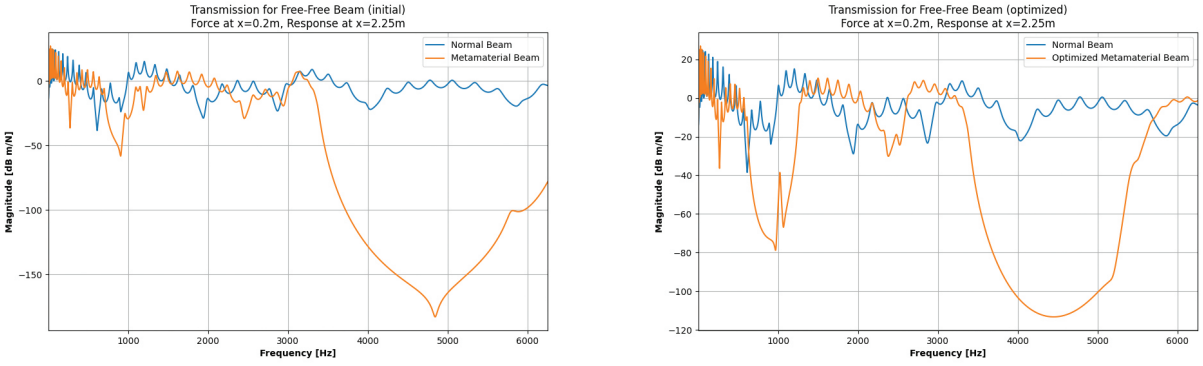


Fig. 5 The initial metamaterial beam dynamic transmission plotted on a linear abscissa in comparison to an equivalent mass control beam can be seen on the left. The right graph shows the transmission for the optimized case where the ~ 4750 Hz frequency bandgap can be seen to reduce in dissipation and shift some of the finite dissipation towards the 1-2 kHz bandgap that starts to form.

in Fig. 6. The control case is plotted above the optimization case. There is a clear reduction in the amplitudes along the entire length of the beam and especially in the peak region of 1-2 kHz.

The optimized geometry itself, a periodic pattern of varying beam thickness (a metamaterial), demonstrates that a Bayesian optimization approach is efficacious. This data-driven method efficiently explored the parameter space and, through a limited number of FEM simulations, identified a configuration that alters the beam's mass and stiffness matrices to create near-ideal bandgap behavior. The GP surrogate model functioned by balancing the exploration of new design regions with the exploitation of promising ones, thereby leading to an optimized final design with fewer trial cases than other schemes, such as neural networks.

Further work needs to explore the extension of this methodology into experimental spaces. The current study was limited to a two-dimensional simulated Timoshenko beam model; a logical next step would be to apply this technique to three-dimensional solid or shell model to capture more complex wave phenomena and predict physical systems. The optimization

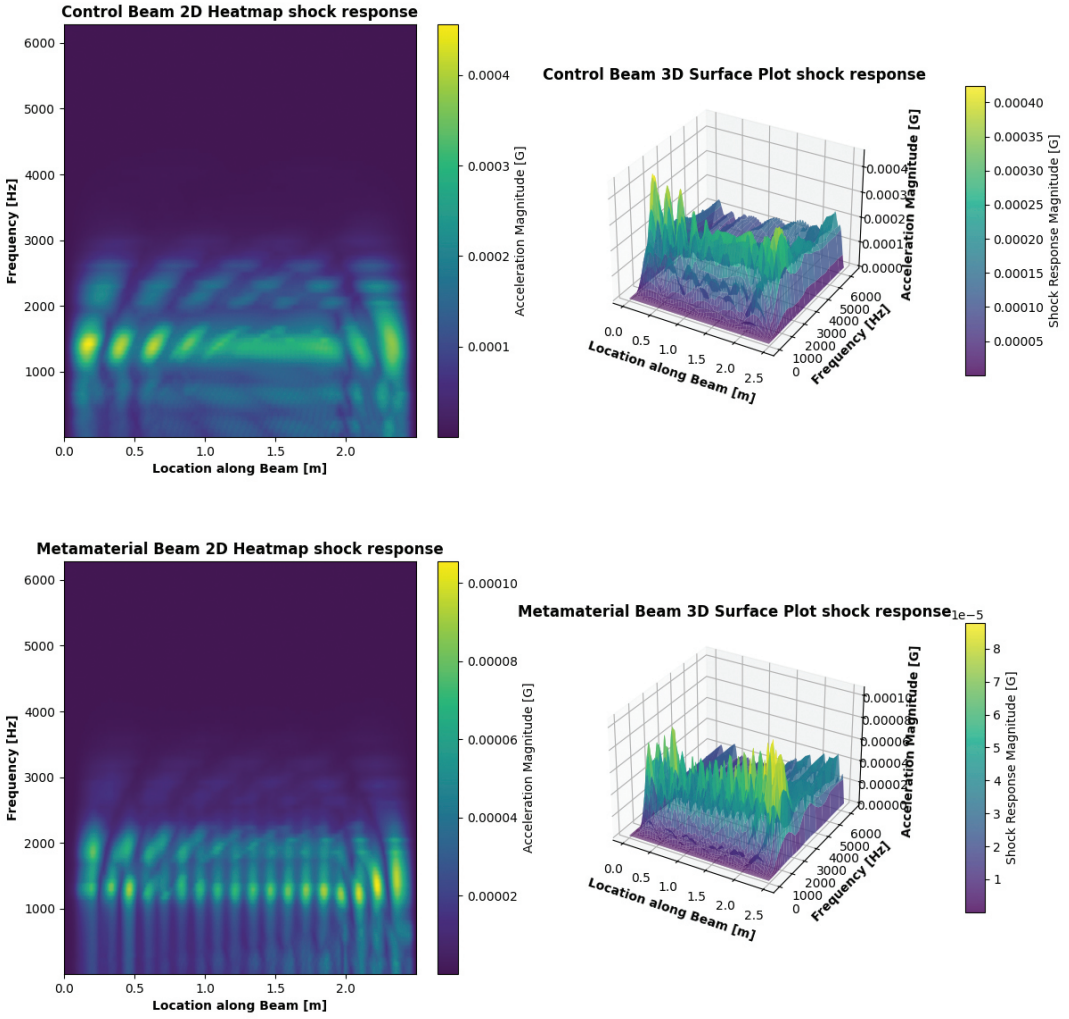


Fig. 6 SRS along the length of the beam for the displacement DOF. The optimized structure shows large reductions in the magnitude of the maximum accelerations for a hypothetical SDOF oscillator in the desired frequency band of 1-2 kHz. The amplitude reduction shows the potential for this scheme to be implemented to tune shock spectra for specific frequency transmission.

scheme can also be extended to a larger parameter space, for example, minimizing both the SRS and the overall structural mass, or by tuning for multiple shock inputs with different spectral characteristics.

Conclusion

This paper presented a novel methodology for shaping an SRS via a geometrically-driven metamaterial beam for enhanced shock response attenuation. By combining a two-dimensional first-order Timoshenko beam finite element model with a Bayesian optimization scheme, we demonstrated a systematic approach to manipulate a beam's geometry to achieve the specific objective of reducing the spectrum in the 1-2 kHz frequency band.

The results unequivocally showed that the optimized metamaterial beam suceeds in attenuating responses relative to a uniform control beam of equivalent mass. The attenuation of the resonant structures is directly attributed to the creation and shifting of structural bandgaps as seen in the transmission plots, which act as a mechanical wave-guide for propagation of structure-borne waves. The success of the optimization, driven by minimizing the area under the PVSS, highlights the efficacy of using a GP-based Bayesian optimization for computationally expensive, black-box problems in structural design.

In summary, this work provides initial simulation results for developing potential passive shock mitigation structures through the reduction of hypothetical SDOF oscillator peak responses.

References

1. Alya Alhammadi, Jin-You Lu, Mahra Almheiri, Fatima Alzaabi, Zineb Matouk, Mohamed Al Teneiji, Rashid K. Abu Al-Rub, Vincenzo Giannini, and Dong-Wook Lee. Numerical modelling and optimization of two-dimensional phononic band gaps in elastic metamaterials with square inclusions. *Applied Sciences*, 11(7), 2021.
2. Mahmoud I. Hussein, Matthew J. Leamy, and Massimo Ruzzene. Dynamics of phononic materials and structures: Historical origins, recent progress, and future outlook. *Applied Mechanics Reviews*, 66(4):040802, 2014.
3. Monty Kennedy. *Correlation of and Development of Procedure to Use a Resonant Plate With Mechanical Excitation for Shock Testing Small-to-Medium Size Spacecraft and Provide Aerospace Shock Analysis and Testing Guidelines*. PhD thesis, Michigan Technological University, 2023. Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2024-05-08.
4. William S. Larsen. *Analysis of the Shock Response Spectrum and Resonant Plate Testing Methods*. PhD thesis, Michigan Technological University, 2018.
5. B. Liu and C. Ji. Automated metamaterial design with computer model emulation and bayesian optimization. *Applied Mechanics and Materials*, 2014.
6. H. Luo, J. Xu, Y. Song, and et al. Machine intelligence in metamaterials design: a review. *Oxford Open Materials Science*, 4(1), 2024.
7. C. Sharpe, C. Seepersad, D. A. Tortorelli, and S. E. Watts. Design of mechanical metamaterials via constrained bayesian optimization. *OSTI*, 2018.
8. Zhichao Tian, Yang Yang, Sui Zhou, Tian Zhou, Ke Deng, Chunlin Ji, Yejun He, and Jun S. Liu. High-dimensional bayesian optimization for metamaterial design. *Materials Genome Engineering Advances*, 2(4):e79, 2024.
9. Thomas Vasileiadis, Jeena Varghese, Visnja Babacic, Jordi Gomis-Bresco, Daniel Navarro Urrios, and Bartlomiej Graczykowski. Progress and perspectives on phononic crystals. *Journal of Applied Physics*, 129(16):160901, 04 2021.
10. Y. Yang, C. Ji, and K. Deng. Rapid design of metamaterials via multitarget bayesian optimization. *The Annals of Applied Statistics*, 15(2), 2021.

Chapter 5
Physics-based Online Cyber-Physical Neural Network Models for Real-time Hybrid Simulation

Faisal Malik, Liang Cao, James Ricles, and Austin Downey

Abstract Real-time hybrid simulation (RTHS) is an experimental testing methodology that divides a structural system into an analytical and an experimental substructure. The analytical and experimental substructures are modeled numerically and physically in the laboratory, respectively. The substructures are kinematically linked at their interface degrees of freedom, and the coupled equations of motion are solved in real-time to obtain the response of the complete system. A key challenge in applying RTHS to large or complex structures is the limited availability of physical devices, which makes it difficult to simultaneously represent all required experimental components in the experimental substructure. The present study addresses this challenge by introducing a Physics-based Online Cyber-Physical Neural Network (OCP-NN) model. The physics-based OCP-NN model leverages real-time data from a single physical device (i.e., the experimental substructure) to replicate its behavior at other locations in the system, thereby significantly reducing the need for multiple physical devices. The proposed method is demonstrated through RTHS of a two-story reinforced concrete frame subjected to seismic excitation and equipped with Banded Rotary Friction Dampers (BRFDs) in each story.

Keywords Banded Rotary Friction Dampers · LSTM Neural Network · Online Model Updating · Physics-based Neural Network · Real-time Hybrid Simulation

Introduction

Real-time hybrid simulation (RTHS), also known as cyber-physical simulation, is an experimental testing methodology that integrates numerical modeling and physical testing to evaluate the dynamic response of structural systems subjected to realistic loading scenarios such as earthquakes, wind, and tsunami. In a RTHS, the structural system is divided into an analytical and an experimental substructure. The analytical substructure is modeled numerically, such as through finite element modeling, while the experimental substructure is physically present in a laboratory. The two substructures are kinematically linked at their interface degrees of freedom (DOF), and the resulting equations of motion are solved in real-time, allowing an accurate evaluation of the system's performance under dynamic loading.

A significant challenge in RTHS occurs when the number of experimental components installed in a structure exceeds the number of physical devices available in the laboratory. To address this limitation a novel online cyber-physical neural

Faisal Malik
Graduate Research Asst., Civil & Environmental Engineering, Lehigh Univ, Bethlehem, PA 18015
e-mail: mamc22@lehigh.edu

Liang Cao
Assistant Professor, University of Mississippi, Oxford, MS.
e-mail: lcao2@olemiss.edu

James Ricles
Professor, ATLSS Engineering Research Center, Lehigh Univ, Bethlehem, PA 18015
e-mail: jmr5@lehigh.edu

Austin Downey
Associate Professor, Mechanical Engineering, University of South Carolina, SC 29208
e-mail: AUSTINDOWNEY@sc.edu

© The Author(s), under exclusive license to River Publishers 2026

Kristin B. Zimmerman, PhD (eds.), *Data Science in Engineering, 2026, Vol. 9*, Conference Proceedings of the Society for Experimental Mechanics Series, https://doi.org/10.13052/rp-9788743814207

network method was developed by Malik et al. [1]. In the framework, one or more experimental substructures of devices are modeled physically and a neural network (NN) model is used to model the remaining devices in the structure. This NN model takes real-time data from the physical device to enhance its prediction accuracy at remaining locations in the structure. The framework was validated on a two-story reinforced concrete (RC) frame equipped with a banded rotary friction damper (BRFD) [2] located at each story and subjected to seismic excitations.

The BRFD features a rotating drum with three steel bands, lined with a ceramic friction material (GGA-Cured, Rigid), wrapped around the drum. The pretension force in the bands is applied through two Tolomatic RSA50 BN02 electric actuators with a stroke of $\pm$ 89 mm. When subjected to external excitation, the rotation of the drum produces an increasing contact pressure profile, which is minimum at the slack end and maximum at the taut end of the bands. As the drum rotates, a frictional torque is produced to counteract the applied force. This phenomenon, known as the self-energizing mechanism, allows the damper to achieve a high friction capacity with minimal energy input. The BRFD has a stroke of 45 mm and a variable force capacity that depends on the applied pretension force.

The RTHS configuration is shown in Figure 1. In the RTHS framework, the structural system is divided into three different substructures: (a) the analytical substructure, which numerically models the frame without the dampers; (b) the experimental substructure, which physically represents the first story damper; and, (c) the OCP-NN model which numerically models the second story damper. The analytical substructure is numerically modeled in HyCOM-3D [3]. The beams and columns are modeled using explicit force-based fiber elements (FBE) with RC sections. The braces are modeled using elastic beam column elements, and the geometric nonlinearities are accounted though a lean-on P$-\Delta$ column.

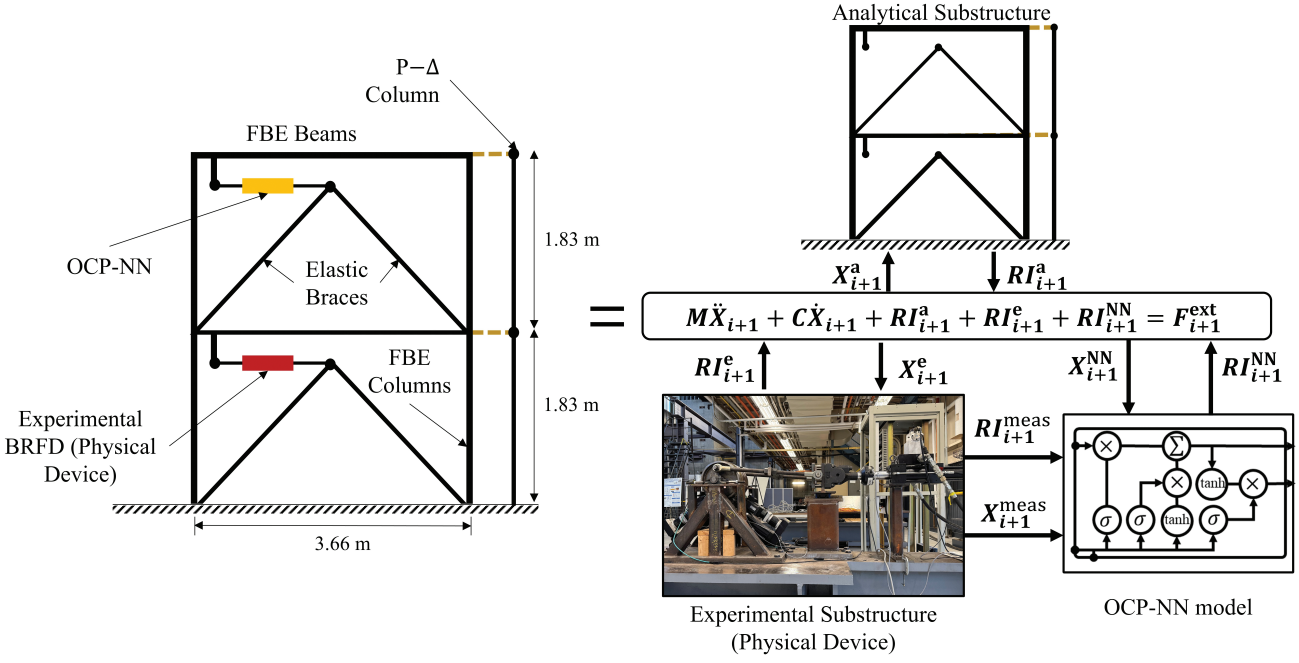


Fig. 1 RTHS configuration with the OCP-NN model

In the RTHS, the model based explicit integration algorithm MRK-α [4] is used to solve the weighted equations of motion. The algorithm determines the displacement and velocity of the current timestep based on the calculated displacement, velocity and acceleration at the previous timestep. Then, the displacement of the analytical substructure $\boldsymbol{X}^{\mathrm{a}}_{i+1}$ is imposed onto the substructure to obtain its restoring force vector $\boldsymbol{RI}^{\mathrm{a}}_{i+1}$. Simultaneously, the damper deformation at the first story is imposed onto the physical device using hydraulic actuators to obtain its restoring force $\boldsymbol{RI}^{\mathrm{e}}_{i+1}$. The measured response of the BRFD, including its measured force, deformation and the tension forces in the two electric actuators along with the deformation at the second story damper are input to the OCP-NN model to obtain the NN's restoring force $\boldsymbol{RI}^{\mathrm{NN}}_{i+1}$. Once the restoring forces from the three substructures are obtained, the global restoring force vector is assembled and the acceleration of the current timestep is completed. The integration algorithm then proceeds to the next timestep. This iterative procedure is repeated until the simulation is completed.

Architecture of the Physics-Based OCP-NN Model

The architecture of the physics-based OCP-NN model is shown in Figure 2. As discussed earlier, the inputs to the physics-based OCP-NN model at each timestep are the measured responses from the physical device, which include the measured force and deformation, and the measured tension forces in the two electric actuators. The physics-based OCP-NN model also uses the deformation at the location of the NN model as its input and predicts the parameters of a numerical model of the device in real-time. These parameters are passed through a physics layer, which is based on the constitutive model of the physical device to predict the force at the location of the physics-based OCP-NN model.

For the device used in the current study i.e., the BRFD the constitutive model is based on the LuGre dry friction model [5]. The output force from the LuGre dry friction model is given by:

$$F = \sigma_0 z + \sigma_1 \dot{z} + \sigma_2 v \tag{1}$$

where v is the velocity at the interface between the drum and the bands, σ_0 is the aggregate bristle stiffness, σ_1 is the microdamping, and σ_2 is the viscous friction. z is an evolutionary variable given by:

$$\dot{z} = v - \sigma_0 \frac{|v|}{g(v)} z \tag{2}$$

where $g(v) = F_c + (F_s - F_c)\, e^{-\left(\frac{v}{v_s}\right)^2}$, F_s is the static friction, F_c is the coulomb friction, and v_s is the stribeck velocity.

For the studied device, F_c and F_s depend on the applied pretension force and are therefore determinate. σ_2 is taken to be zero as the friction force is not dependent on velocity. Therefore, the physics-based OCP-NN model predicts the two parameters σ_0 and σ_1 in real-time.

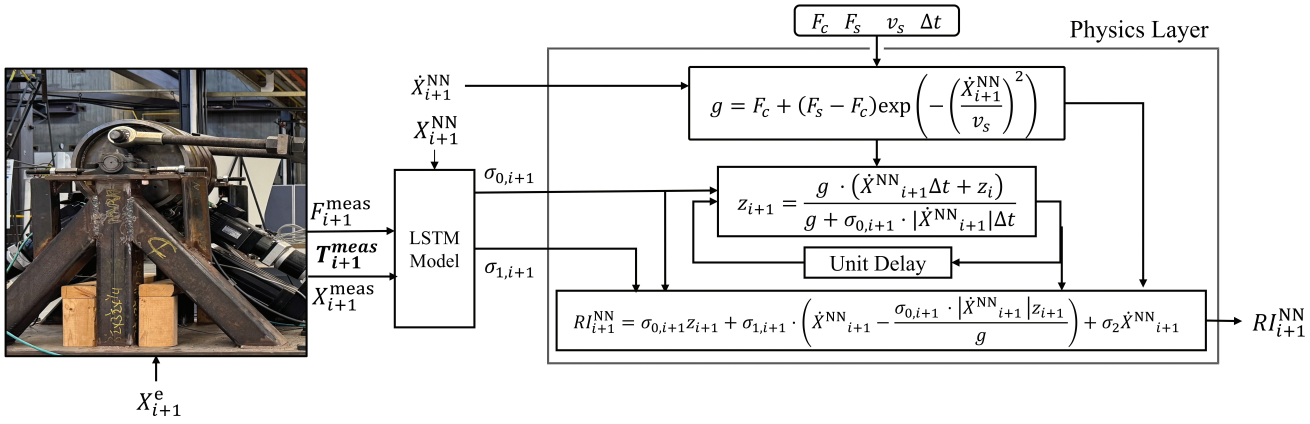


Fig. 2 Architecture of the Physics-based OCP-NN model

The architecture of the LSTM model used in the physics-based OCP-NN model is shown in Figure 3. The measured responses from the physical device and the input deformation of the OCP-NN model are input to two LSTM layers with 16 neurons each to obtain a higher dimensional representation of the inputs. The output from these two layers is concatenated and passed through two consecutive LSTM layers of 48 neurons each. The output from the final LSTM layer is then passed through a dense layer to get the two output quantities σ_0 and σ_1 at each timestep. The physics-based OCP-NN model is trained using the Adam optimizer with an initial learning rate of 0.001, which is decayed by 2% every 150 epochs. A batch size of five is used for training the model. It is important to regularize the NN model for applications in RTHS [6]. Therefore, a dropout of 10% after each layer is used as the regularization method.

To generate the training data, a suite of ground motion records is obtained from the PEER NGA West2 database and numerical simulations of the RC frame are performed using a numerical model of the BRFD (i.e., the LuGre model) to obtain the damper deformations at the first and second story. These deformations are subsequently imposed on the physical device through hydraulic actuators to obtain the corresponding measured forces. These force deformation pairs are then assembled to form the training and validation dataset.

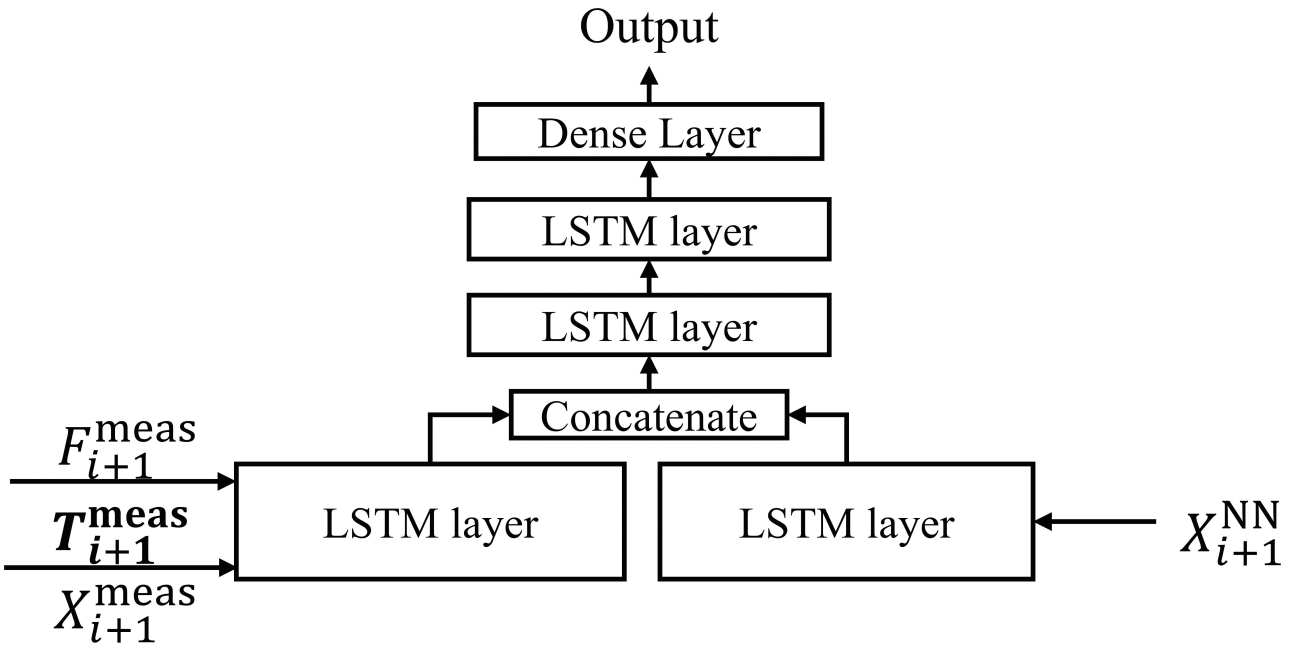


Fig. 3 Architecture of the LSTM model utilized in the physics-based OCP-NN model

Validation of the Physics-Based OCP-NN Model

This section presents the results from the validation of the physics-based OCP-NN model on harmonic excitation and seismic excitations. Figure 4 presents a comparison between the measured and predicted force of the BRFD under a harmonic excitation with an amplitude of 16.5 mm and a frequency of 0.5 Hz. As shown in the figure, the predicted response matches the measured data well, yielding a normalized root mean square error (NRMSE) of 3.72%. Beyond demonstrating accurate prediction capability, the physics-based OCP-NN model also provides enhanced interpretability of the system behavior. In particular, the model captures the characteristic *backlash* phenomenon in the BRFD response, which manifests as a stiffness reduction during load reversal.

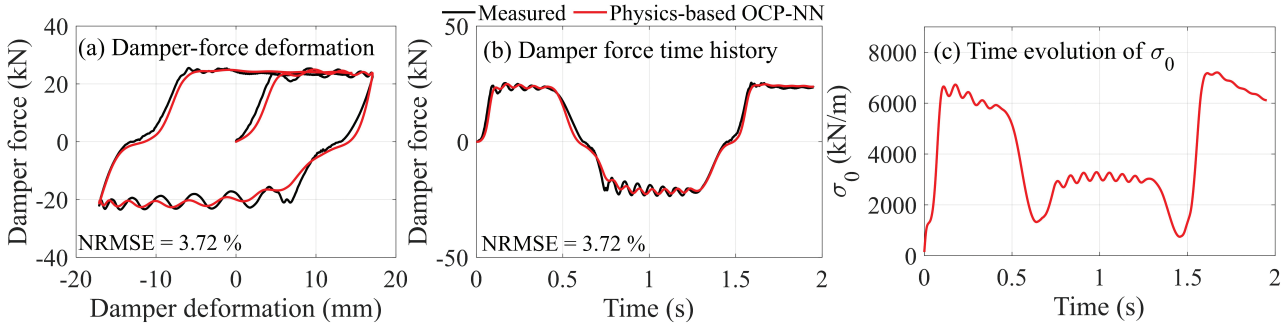


Fig. 4 Comparison of BRFD measured and predicted force for a harmonic excitation of amplitude 16.5 mm and frequency 0.5 Hz: (a) force-deformation; (b) force time history; (c) time evolution of σ_0.

This phenomenon is explained through the variation of the parameter, σ_0 (see Figure 4(c)). Specifically, σ_0 decreases during the de-energizing phase and subsequently increases during the re-energizing phase, thereby providing a clear physical basis for the observed hysteretic behavior.

For validating the physics-based OCP-NN model under seismic excitations, four earthquake ground motions were selected. Figure 5 presents a comparison between the measured and predicted BRFD force for the Kobe ground motion scaled

to the design basis earthquake (DBE) hazard level. As shown in Figure 5(a), the physics-based OCP-NN model accurately reproduces the response of the BRFD, with a normalized root mean square error (NRMSE) of 1.88%, thereby demonstrating its strong predictive capability. Moreover, the time evolution of σ_0 (Figure 5(b)) provides additional physical insight into the underlying mechanism. In particular, σ_0 does not remain constant but evolves continuously due to the interaction between the bands and the drums, highlighting the model's ability to link observed behavior with interpretable physical parameters.

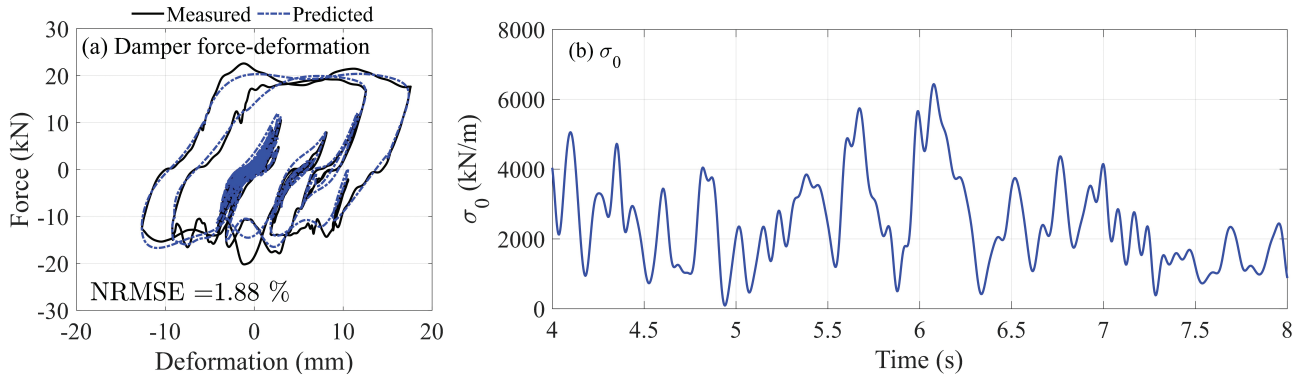


Fig. 5 Comparison of measured and predicted force for Kobe earthquake recorded at Kobe University: (a) force-deformation, and (b) time evolution of σ_0 between t = 4 and t = 8 sec

The mean μ and standard deviation (σ) of error quantities based on NRMSE, mean absolute error (MAE), R^2, and time response assurance criteria (TRAC) for the validation set are summarized in Table 1. The mean NRMSE of 2.61% with a relatively small standard deviation of 0.62% indicates that the predicted forces closely match the measured responses across different validation cases, with minimal variability. Similarly, the MAE with a mean of 0.76 and a standard deviation of 0.19 confirms that the model achieves low pointwise discrepancies in reproducing the force–deformation response. The mean and standard deviation for the R^2 value is 0.95 and 0.04, respectively. R^2 values close to unity further validates the NN model's ability to capture the overall response trends. Importantly, the TRAC values achieve a mean of 0.97 with a narrow variability of 0.02, highlighting the capability of the OCP-NN model to preserve the temporal characteristics.

Table 1 Mean and standard deviation of error metrics on the validation dataset

Metric	NRMSE (%)	MAE	R^2	TRAC
$\mu \pm \sigma$	2.61 ± 0.62	0.76 ± 0.19	0.95 ± 0.04	0.97 ± 0.02

RTHS Results

The Northern Calif-03 ground motion recorded at the Ferndale City Hall station and scaled to the DBE hazard level was used in the RTHS. The corresponding roof displacement and second-story damper deformation responses are shown in Figure 6, while the moment–curvature relationships at the base of the first-story column and at the end of the first-floor beam are presented in Figure 7. As can be observed, the structure undergoes pronounced nonlinear behavior during the excitation. In particular, the moment–curvature plots (Figure 7) clearly exhibit nonlinear hysteretic response, indicating the development of inelastic deformations in both the column and the beam. Moreover, residual roof displacement is evident in Figure 6(a), highlighting the accumulation of permanent deformation in the global system.

To further assess the accuracy of the physics-based OCP-NN model, the second-story damper deformation obtained during the RTHS was re-imposed on the physical device following the RTHS to record the corresponding measured force. Figure 8 shows a comparison between the measured and predicted force–deformation responses. As illustrated, the predicted response closely follows the measured behavior, achieving a normalized root mean square error (NRMSE) of 4.84%. This strong agreement highlights the capability of the OCP-NN model to generalize beyond the real-time simulation and accurately reproduce the nonlinear response characteristics of the BRFD under imposed deformations.

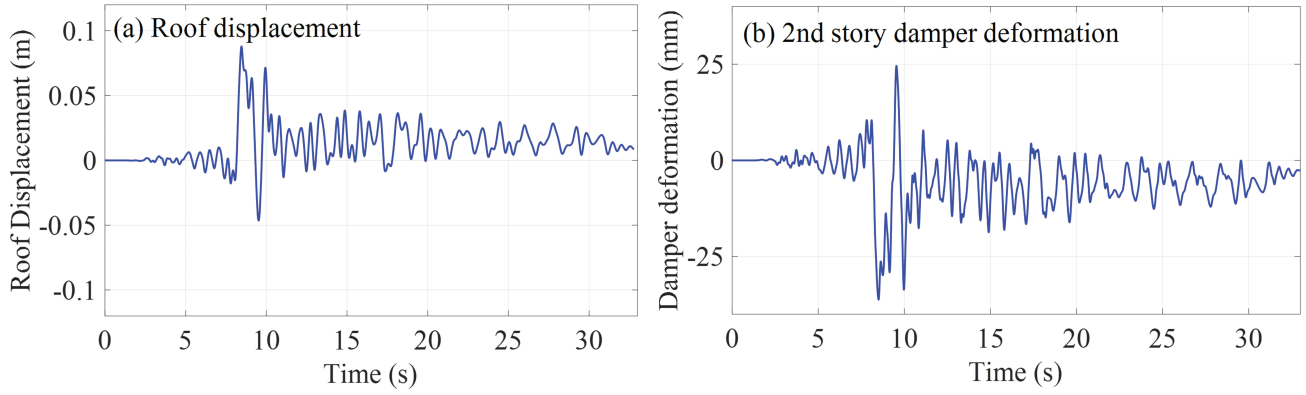


Fig. 6 Results from the RTHS of Northern Calif-03 ground motion: (a) Roof displacement, and (b) second story damper deformation

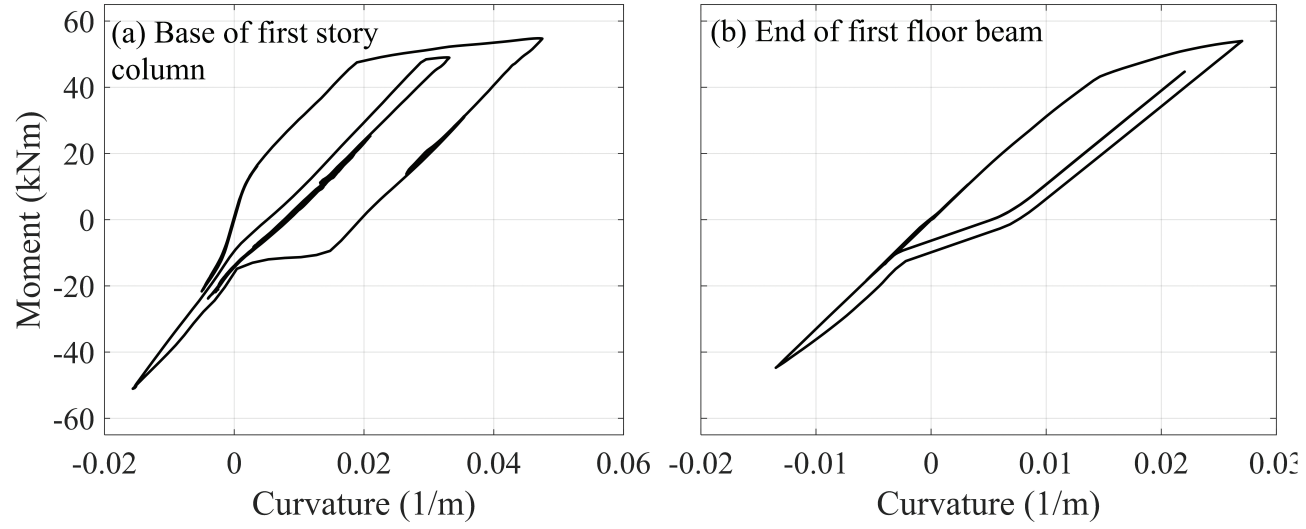


Fig. 7 Moment curvature obtained from the RTHS of Northern Calif-03 ground motion: (a) Base of first story column, and (b) end of first floor beam

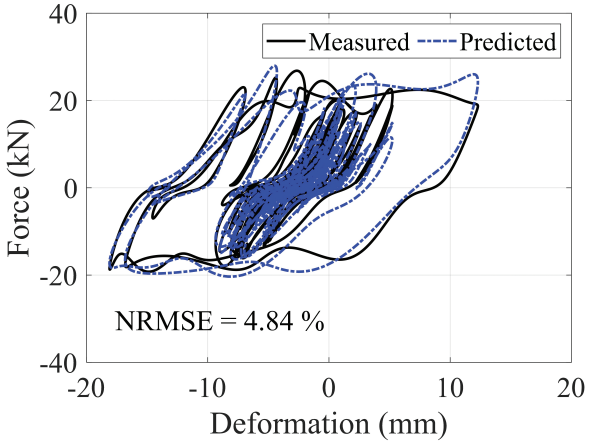


Fig. 8 Comparison of the measured and predicted force deformation during the RTHS of the Northern Calif-03 ground motion

Summary and Conclusions

This study presented a novel Physics-based Online Cyber-Physical Neural Network (OCP-NN) model for application in real-time hybrid simulation (RTHS) of structural and mechanical systems. The OCP-NN framework addresses the challenge of limited availability of physical devices in a RTHS by leveraging real-time data from a single physical device to replicate the behavior of additional devices at other locations in the system. The approach was demonstrated through the RTHS of a two-story reinforced concrete (RC) frame equipped with Banded Rotary Friction Dampers (BRFDs) in each story and subjected to seismic excitations.

The OCP-NN integrates a long short-term memory (LSTM) neural network with a physics-based constitutive model of the BRFD that is based on the LuGre dry friction formulation. The network predicts key parameters (σ_0 and σ_1) in real time, which are passed through a physics layer to compute the damper restoring forces.

Validation results showed strong predictive performance of the physics-based OCP-NN model, with low error quantities across multiple ground motions. For harmonic excitation, the model achieved an NRMSE of 3.72%, while for the Kobe ground motion scaled to the DBE hazard level, the error reduced to 1.88%. Across the validation dataset, the mean and standard deviation for the NRMSE, MAE, R^2, and TRAC were $2.61\% \pm 0.62\%$, 0.76 ± 0.19, 0.95 ± 0.04, and 0.97 ± 0.02, respectively. The results confirmed both high accuracy and robustness of the proposed approach. Importantly, the time evolution of σ_0 provided physical interpretability by capturing backlash effects and band–drum interactions.

The RTHS of the RC frame using the Northern Calif-03 ground motion further validated the framework. The system exhibited pronounced nonlinear and hysteretic behavior, including residual roof displacement and inelastic moment–curvature response in the columns and beams of the structure. When the second-story damper deformation obtained during the RTHS was re-imposed on the physical device, the OCP-NN reproduced a measured force–deformation response with an NRMSE of 4.84%, demonstrating its ability to generalize beyond the real-time simulation.

Acknowledgments This material is based upon work supported by the National Science Foundation (NSF) under Grant Nos. CCF-2234921, and CMMI-1463497, and from MTS Corporation. The experiments reported herein were performed at the NHERI Lehigh Experimental Facility, whose operation is supported by a grant from the NSF under Cooperative Agreement No. CMMI-2037771. The authors are grateful for the support of the sponsors and acknowledge the effort of the NHERI Lehigh Experimental Facility staff. The opinions, findings, conclusions and recommendations expressed herein are those of the authors and do not necessarily reflect the views of the NSF or the other sponsors.

References

1. F. N. Malik, L. Cao, J. Ricles and A. Downey, “Online Cyber-Physical Neural Network Model for Real-Time Hybrid Simulation,” *Earthquake Engineering & Structural Dynamics* , **54**(15): 3457–3474, 2025.
2. A. Downey, L. L. S. Cao, D. Taylor and J. Ricles, "High Capacity Variable Friction Damper Based on Band Brake Technology," *Engineering Structures*, **113**: 287-298, 2016.
3. J. Ricles, C. Kolay and T. Marullo, "HyCoM-3D: A Program for 3D Multi-Hazard Nonlinear Analysis and Real-Time Hybrid Simulation of Civil Infrastructure Systems," ATLSS Engineering Research Center, Report No. 20-02, Lehigh University, Bethlehem, 2020.
4. C. Kolay and J. Ricles, "Improved Explicit Integration Algorithms for Structural Dynamic Analysis with Unconditional Stability and Controllable Numerical Dissipation," *Journal of Earthquake Engineering*, **23(**5): 771-792, 2017.
5. C. Canudas-de-Wit, H. Olsson, K. Åström and P. Lischinsky, "A new model for control of systems with friction," *IEEE Transactions on Automatic Control*, **40**(3): 419 - 425, 1995.
6. F. N. Malik, D. N. Gorini, J. Ricles and M. Rahnemoonfar, "Multi-Physics Framework for Seismic Real-Time Hybrid Simulation of Soil–Foundation–Structural Systems," *Engineering Structures*, **334**: 120247, 2025.

Chapter 6
Physics-Informed Machine Learning Part III: Hard-Constraint ODE Method for Structural Dynamics

Mohsen Gol Zardian, Austin R.J. Downey, Eleonora Maria Tronci, Conor Madden, Daniel Coble, Sina Navidi, and Chao Hu

Abstract Physics-informed machine learning (PIML) is a methodology that combines principles from physics with machine learning (ML) techniques to enhance the accuracy and interpretability of predictive models. By incorporating physical laws and constraints into the learning process, physics-informed machine learning enables more robust predictions and reduces the need for large amounts of training data. PIML has a wide range of applications in science and engineering, such as modeling physical systems, solving partial differential equations, and performing inverse analysis and optimization.

In Part III of this series, the authors provide an overview of a hard-constraint, indirect-measurement PIML approach for structural dynamics. An ordinary differential equation (ODE) solver implemented within the automatic differentiation engine serves as a differentiable layer that enforces the mass-damping-stiffness equilibrium at every inference step. A compact neural network ingests a window of past acceleration samples and outputs an effective stiffness. The embedded Runge-Kutta-4 integrator then propagates states so that the predicted acceleration satisfies the equation of motion. The technique is demonstrated on a deliberately simple numerical example: a simulated spring, mass, and dashpot whose stiffness drifts over time along several representative paths. A collection of runs is generated, some used for training, the rest held back for validation so that the model must generalize to unseen stiffness trajectories. Performance is benchmarked against two reference solvers. First, a physics-only baseline consisting of a conventional ODE integrator that assumes the stiffness is fixed and therefore cannot reflect its time variation. Second, a data-only baseline consisting of a neural network of identical size trained to minimize next-step acceleration error, without any governing-equation guidance. Results show that the hard-constraint PIML model approaches the data-only network in response accuracy while offering far greater physical interpretability through its explicit stiffness estimates. The paper reports on hyper-parameter tuning for the multi-layer perceptron so readers can readily

Mohsen Gol Zardian · Conor Madden · Daniel Coble
Department of Mechanical Engineering
University of South Carolina, Columbia, SC 29208
e-mail: golzardm@email.sc.edu; CMMADDEN@email.sc.edu; daniel.coble@duke.edu

Austin R.J. Downey
Department of Civil and Environmental Engineering
University of South Carolina, Columbia, SC 29208
e-mail: austindowney@sc.edu

Eleonora Maria Tronci
Department of Civil and Environmental Engineering
New York University
e-mail: e.tronci@northeastern.edu

Daniel Coble
Department of Mechanical Engineering & Materials Science
Duke University
e-mail: daniel.coble@duke.edu

Sina Navidi · Chao Hu
School of Mechanical, Aerospace, and Manufacturing Engineering
University of Connecticut
e-mail: sina.navidi@uconn.edu; chao.hu@uconn.edu

© The Author(s), under exclusive license to River Publishers 2026

Kristin B. Zimmerman, PhD (eds.), *Data Science in Engineering, 2026, Vol. 9*, Conference Proceedings of the Society for Experimental Mechanics Series, https://doi.org/10.13052/rp-9788743814207

adapt the hard-constraint indirect-measurement PIML method to their own inverse-problem applications. Project code are artifacts available are through thorough a public repository.

Keywords Physics-informed · physics-based model · data driven model · machine learning · stiffness identification

Introduction

The rapid growth of sensing technologies and advances in machine learning (ML) have opened up new possibilities for modeling and predicting how engineering systems behave. In structural dynamics and health monitoring, data-driven models have proven highly successful by learning patterns directly from large datasets [1]. However, when it comes to systems such as those encountered in structural dynamics, purely data-driven approaches face two major challenges: they often require far more data than is practical to collect, and their predictions can sometimes violate the relevant physics [2]. These limitations have led to the emergence of physics-informed machine learning (PIML) [3, 4], an approach that blends data-driven modeling with the governing equations [5] of the system to deliver both accurate and physically meaningful predictions of a structures repose on state [6].

Recent work by the authors has explored different ways to integrate physical knowledge into ML models. In Part I of this series, they proposed how embedding domain equations can enhance robustness, improve interpretability, and boost predictive performance in engineering applications [7]. In Part II, they demonstrated the practical application of PIML techniques to structural dynamics, showing how physics-informed architectures can forecast system responses [8]. Their work illustrates the advantages of integrating physical models with neural networks for the inverse challenge of system parameter estimation.

In many fields, data–driven models have achieved remarkable success by capturing patterns directly from large datasets. However, for physical systems such as structural dynamics, purely data–driven approaches face two important limitations: they often require more data than is practical to collect, and they may generate predictions that are inconsistent with known physical laws. This mismatch has motivated the development of physics–informed machine learning (PIML), which integrates governing equations into ML training to achieve both accuracy and interpretability [9].

PIML has already demonstrated its value in applications ranging from fluid mechanics to materials science. In these areas, physical constraints can be incorporated in two main ways. In soft–constraint approaches, the residual of the governing equations is added to the loss function, alongside the mismatch between real and predicted states (As discussed in Part IV of this series [10]). By contrast, hard–constraint approaches embed the equations directly into the model architecture, ensuring that physical laws are satisfied exactly during both training and validation [11]. While harder to design, this strategy is especially important in domains where reliability and physical consistency are critical.

The rapid progress of differentiable solvers has made these ideas more practical. Neural ordinary differential equations (Neural ODEs) [12] allow parameters of dynamical systems to be learned by backpropagating through numerical integrators, while more recent methods have improved stability and efficiency. At the same time, learning frameworks such as DeepONet [13] and Fourier Neural Operators (FNOs) [14] have emerged as powerful tools for approximating families of differential equation solutions. Together, these advances show that combining learning with physics can reduce data requirements, improve generalization, and yield models that are more trustworthy for engineering applications.

Structural dynamics presents a particularly compelling case for PIML. Systems such as bridges, buildings, and aerospace structures are characterized by time–varying properties, nonlinearities, and environmental influences that are difficult to capture with purely physics–based models. At the same time, safety considerations demand models that remain interpretable and physically consistent. Reviews in this area highlight both the potential and the challenges of PIML, noting that soft–constraint formulations often require delicate tuning and may drift from the governing laws when extrapolating [3, 4]. Hard–constraint approaches, by directly encoding the equations of motion, provide a more principled path that ensures predicted responses always respect the underlying mechanics.

In this paper, we focus on the development of a hard–constraint PIML framework for structural dynamics. The method is demonstrated on a mass, spring, and damper system with time-varying stiffness [15], chosen as a benchmark problem to illustrate the challenges of parameter identification under nonstationary conditions. Our framework combines a neural network that estimates effective stiffness from short windows of measured accelerations with a physics layer that propagates states through a numerical integrator. The training objective is defined solely by the mismatch between predicted and observed accelerations, eliminating the need for additional residual penalties. By construction, the governing equation is satisfied at every time step, yielding predictions that are not only accurate but also physically interpretable. We compare the proposed model against physics baselines, demonstrating its advantages in accuracy and robustness. Finally, we discuss sensitivity to hyperparameters to evaluate performance and highlight the broader implications of this approach for structural

health monitoring and other engineering applications where physical consistency is critical. Project code are artifacts are through thorough a public repository for this paper[16].

Methodology

This work develops a hard-constraint physics-informed machine learning (PIML) framework to identify and predict the dynamic response of a mass, spring, and damper system. The central idea is to combine a data-driven neural network with a physics layer such that the governing equation of motion is strictly enforced at every time step. Unlike soft-constraint approaches, where the residual of the governing ODE is included as a penalty term in the loss function, the present framework encodes the ODE directly into the model architecture. As a result, the learned solution is more likely to satisfy the physical law, and training is guided solely by data loss.

Data preparation

The dataset comprises trajectories of displacement, velocity, acceleration, stiffness, and external forcing, obtained from simulated or experimental measurements. Each trajectory is stored in tabular format, where rows correspond to discrete time steps and columns correspond to the measured physical quantities. For clarity, let time be denoted by t, displacement by $x(t)$, velocity by $v(t)$, acceleration by $a(t)$, stiffness by $k(t)$, and external force by $F(t)$.

To reduce redundancy while preserving the essential dynamics, all data is passed through the reprocessing pipeline detailed in Algorithm 1 (summarized in Fig. 1) and discussed in what follows. Trajectories are downsampled by a factor s. A sliding window of length W is then applied to extract local temporal patterns from the acceleration histories. To ensure that every training sample encodes both short- and long-horizon information, consecutive windows are grouped into sequences of length T. To promote generalization, the resulting sequences are randomly shuffled and divided into training and validation sets using a prescribed split ratio ρ. Finally, data is arranged into mini-batches of size B, yielding tensors with dimensions (B, T, W), where B is the batch size, T the number of windows per segment, and W the number of time steps within each window. This organization facilitates efficient parallel training while maintaining sufficient temporal resolution. The overall data preparation process used in this study is illustrated in Fig. 1, which outlines the preprocessing pipeline employed in Algorithm 1 to transform raw sensor trajectories into mini-batches.

Algorithm 1 preprocessing pipeline from raw data to mini-batches

procedure BUILDBATCHES($\{\mathrm{CSV}_n\}_{n=1}^{N}$, s, T, W, B, ρ)
Require: each CSV_n has L rows and D columns (e.g., t, x, v, a, k, F)
Ensure: iterable of train/validation mini-batches with shape (B, T, W)
    counts/params: $N \to$ number of trajectories; $L \to$ rows/trajectory; $D \to$ columns; $T \to$ segment length; $W \to$ window length; $B \to$ batch size
    shape $N \times L \times D$**:** $X \leftarrow \mathrm{Stack}(\mathrm{CSV}_1, \ldots, \mathrm{CSV}_N)$
    downsample by factor s**:** $\tilde{X} \leftarrow \mathrm{Downsample}(X, s)$; $\tilde{L} \leftarrow \mathrm{ceil}(L/s)$
    for $n = 1$ **to** N **do**
        for $t = W$ **to** $\tilde{L}$ **do**
            window length W**:** $a_{t:W}^{(n)} \leftarrow [a_{t-W+1}^{(n)}, \ldots, a_{t}^{(n)}]$
        end for
        for $\tau = 1$ **to** $\tilde{L} - T + 1$ **do**
            segment size $T \times W$**:** $S_{\tau}^{(n)} \leftarrow (a_{\tau:W}^{(n)}, \ldots, a_{\tau+T-1:W}^{(n)})$
        end for
    end for
    total segments: $N_{\mathrm{seg}} \leftarrow N(\tilde{L} - T + 1)$
    split ratio ρ**:** $\{\mathrm{train}\}, \{\mathrm{val}\} \leftarrow \mathrm{Split}(\{S_{\tau}^{(n)}\}, \rho)$
    tensor shape (B, T, W)**:** batches $\leftarrow \mathrm{Batch}(\{\mathrm{train/val}\}, B)$
end procedure

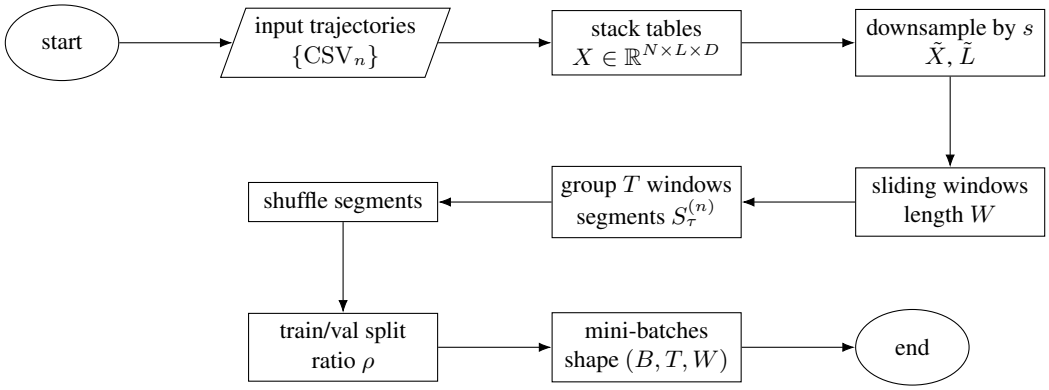


Fig. 1 Preprocessing pipeline from raw trajectories to mini-batches.

Network structure and physics layer

The proposed architecture is designed around the principle of hard-constraint PIML, where the governing physics is enforced directly in the computational structure. As illustrated schematically in Fig. 2, the framework consists of two interconnected components, including a data-driven neural network for parameter estimation and a physics layer that integrates the equations of motion. These components present a hybrid system that combines the flexibility of machine learning with the rigor of classical mechanics.

The data-driven component is a multilayer perceptron (MLP) that receives the windowed acceleration histories of length W as input. The output of this MLP is a variable value of stiffness, $\hat{k}(t)$, which can change across time steps and trajectories. This flexibility enables the model to capture both constant and nonstationary stiffness patterns that may arise from material variability, nonlinearities, or environmental influences.

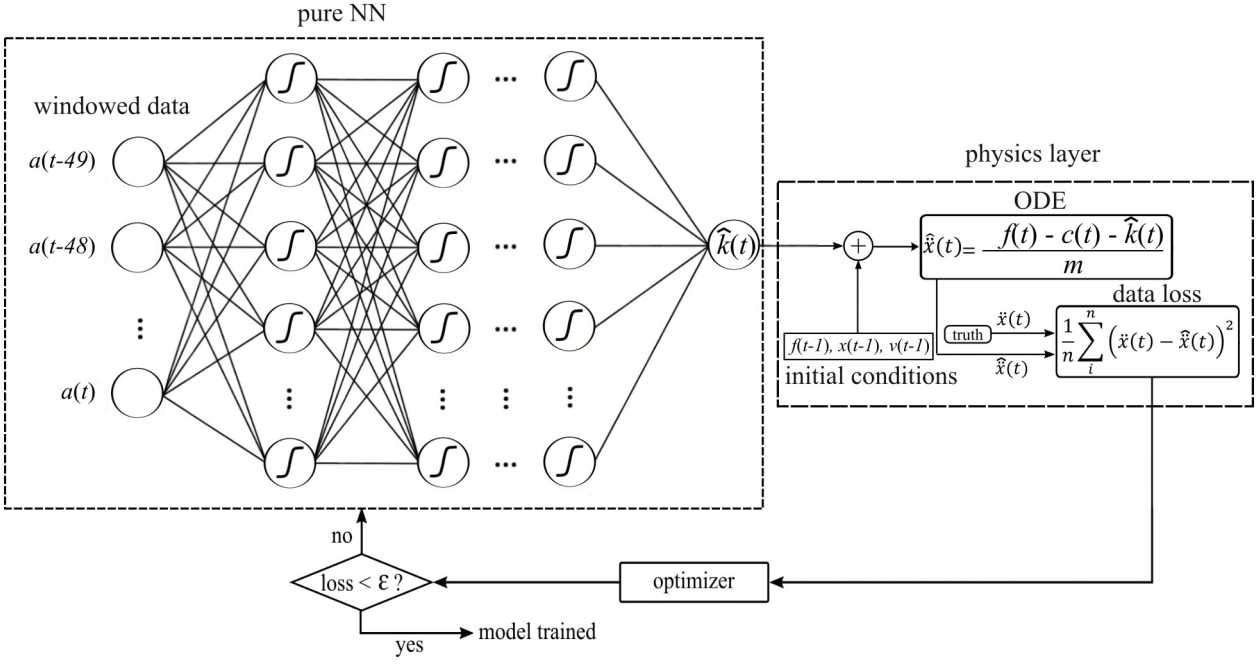


Fig. 2 High-level architecture of the proposed hard-constraint PIML framework. Unlike soft-constraint layouts [9], the governing ODE is directly embedded in the physics layer.

The estimated stiffnesses are then passed to the physics layer, which incorporates the mass, spring, and damper of the motion equation :

$$m\ddot{x} + c\dot{x} + k_{\text{true}}x = F(t) \tag{1}$$

Here, m and c are fixed system parameters, while $\hat{k}(t)$ is provided by the neural network. Given initial conditions and the external force history, the physics layer propagates the system state vector (x, v) forward in time. Numerical integration schemes, such as the fourth-order Runge–Kutta method, are employed to approximate the continuous dynamics with high accuracy. From the updated states, the predicted acceleration $\hat{a}(t)$ is obtained at each time step.

By embedding the ODE solver into the model, the framework ensures that predictions remain consistent with the governing law of motion. Unlike soft-constraint approaches, there is no need to include the residual of the ODE in the loss function, because the residual is automatically satisfied by construction. This design eliminates the need to balance competing loss terms and avoids issues of weighting between data and physics. As a result, the network focuses on learning to map from observed histories to stiffness estimates, while the physics layer guarantees that state updates respect fundamental dynamics. The overall structure achieves a strict hard-constraint formulation that combines physical validity and data-driven adaptability.

To further clarify the inference workflow, Fig. 3 presents the sequential interaction between the neural network and the embedded ODE solver. At each time step t_i, the ML model produces a stiffness estimate $\hat{k}_i$, which is then passed to the ODE solver (here chosen as RK4) together with the external forcing input and the previous system state. The solver propagates displacement and velocity states to the next time step, from which the predicted acceleration is reconstructed. This recursive process enforces the governing dynamics at every step while allowing the network to compute the stiffness evolution from observed acceleration windows.

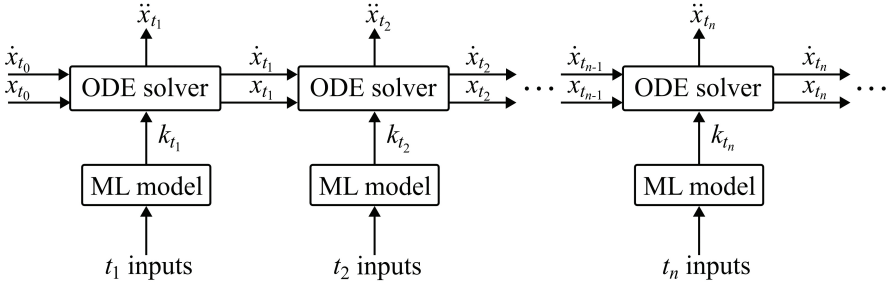


Fig. 3 Detailed workflow of the hard-constraint inference method. The ODE solver is embedded within the automatic differentiation engine, and the ML model generates time-varying stiffness estimates $\hat{k}(t)$ at each step.

Training procedure

The training strategy follows the hard-constraint principle by optimizing only for consistency between predicted and measured accelerations. The training pipeline is summarized in Algorithm 2. At each epoch, the training set is divided into mini-batches, and for each batch the MLP produces variable stiffness $\hat{k}(t)$. These estimates are passed to the physics layer, which integrates the mass, spring, and damper in the equation of motion to propagate displacement and velocity states and compute predicted accelerations $\hat{a}(t)$. The model output is then compared directly with measured accelerations $a(t)$ to compute the loss.

The loss function is defined exclusively as the mean squared error (MSE):

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left(\ddot{x}(t)_i - \hat{\ddot{x}}(t)_i\right)^2, \tag{2}$$

where N is the number of predicted time steps within a batch. This simple formulation reflects the hard-constraint design, since the governing ODE is embedded in the physics layer, the residual is inherently satisfied and does not need to be included in the loss function. The training objective, therefore, focuses entirely on reducing the difference between observed and predicted accelerations.

Algorithm 2 Physics-informed training with time-distributed MLP

procedure TRAINPINN($\mathcal{B}$, E, η; g_θ, $\mathcal{P}_\phi$, m, c)
Require: batches $\mathcal{B}$ of shape (B, T, W); forces $\{F_t\}$; measured accelerations $\{a_t\}$
Ensure: trained parameters θ
  model: $g_\theta : \mathbb{R}^W \to \mathbb{R}$ predicts $\hat{k}_t$; $\mathcal{P}_\phi$ is a mass–spring–damper step (fixed m, c, e.g., RK4)
  number of batches: $M \to |\mathcal{B}|$
  for $e = 1$ **to** E **do**
    for $j = 1$ **to** M **do**
      select batch: $\mathbf{B} \to \mathcal{B}[j]$
      segment index (parallelizable):
      for $i = 1$ **to** B **do**
        for $t = 1$ **to** T **do**
          stiffness per time step: $\hat{k}_t^{(i)} \leftarrow g_\theta(a_{t:W}^{(i)})$
        end for
        initial state: $s_0^{(i)} \leftarrow (x_0^{(i)}, v_0^{(i)})$
        for $t = 1$ **to** T **do**
          forward physics step: $s_t^{(i)} \leftarrow \mathcal{P}_\phi(s_{t-1}^{(i)}, \hat{k}_t^{(i)}, F_t^{(i)})$
          predicted acceleration: $\hat{a}_t^{(i)} \leftarrow \mathrm{Accel}(s_t^{(i)})$
        end for
        segment loss (MSE): $L_{\mathrm{seg}}^{(i)} \leftarrow \frac{1}{T}\sum_{t=1}^{T}(\hat{a}_t^{(i)} - a_t^{(i)})^2$
      end for
      batch loss: $L_{\mathrm{batch}} \leftarrow \frac{1}{B}\sum_{i=1}^{B} L_{\mathrm{seg}}^{(i)}$
      Adam update: $\theta \leftarrow \mathrm{Adam}(\theta, \nabla_\theta L_{\mathrm{batch}}, \eta)$
    end for
  end for
  return θ
end procedure

Model parameters are updated using the Adam optimizer, which provides adaptive learning rates and robust convergence for nonlinear problems. In practice, training stability is further enhanced by techniques such as early stopping and learning rate scheduling. We applied an early stopping criterion during network training to avoid unnecessary computation once the model had converged. The validation loss was monitored at each epoch, and the training process was terminated when no improvement was detected beyond a specified patience threshold. Learning rate scheduling, such as reducing the rate when validation loss plateaus, so the optimizer explores the parameter space efficiently for converging to a stable solution.

The batch-wise structure of the algorithm ensures efficient use of computational resources and supports parallelization across samples. Each batch loss is computed as the average of segment losses, which allows the model to learn stiffness patterns consistently across multiple sequences. By iterating over all batches and epochs, the model gradually refines its parameters until the predicted accelerations closely align with experimental observations. This design guarantees a balance between data-driven adaptation and strict adherence to the underlying physics.

Case Study

To evaluate the proposed hard-constraint PIML framework, we apply it to a classical single-degree-of-freedom (SDOF) spring, mass and damper oscillator subjected to an external force $F(t)$. This canonical model is representative of a wide range of structural dynamics problems in mechanical and civil engineering [15]. The system dynamics are governed by the equation of motion already introduced in Eq. (1), where the objective is to recover the time-varying stiffness parameter $k(t)$ indirectly from measured acceleration responses. The reference physics-based model was implemented in MATLAB Simulink/Simscape and is available in the public repository for this work [16]. The main physical parameters used in the simulation, including the mass, damping coefficient, stiffness range, and excitation characteristics, are summarized in Table 1.

Table 1 System parameters used in the SDOF spring–mass–damper simulation.

Parameter	Symbol	Value
Mass	m	$3.8\ \mathrm{kg}$
Stiffness (maximum)	$k_{\max}$	$1500\ \mathrm{N/m}$
Stiffness (minimum)	$k_{\min}$	$500\ \mathrm{N/m}$
Damping coefficient	c	$0.48\ \mathrm{N/(m \cdot s)}$
External force amplitude	F_0	$1\ \mathrm{kN}$
Excitation frequency	f	$100\ \mathrm{Hz}$

Figure 4 illustrates the two configurations considered in this study. In Fig. 4(a), the assumed oscillator for the system (i.e. the one used in the ODE in Figure 2) is modeled with a conventional linear mass, spring, and damper arrangement in which stiffness $k(t)$ is permitted to vary with time, thereby simulating gradual stiffness degradation or environmental variability. Fig. 4(b) shows the system that is used to develop the dataset; an additional nonlinear restoring force f is introduced in parallel with the damper, enabling the system to capture the more complex friction behavior.

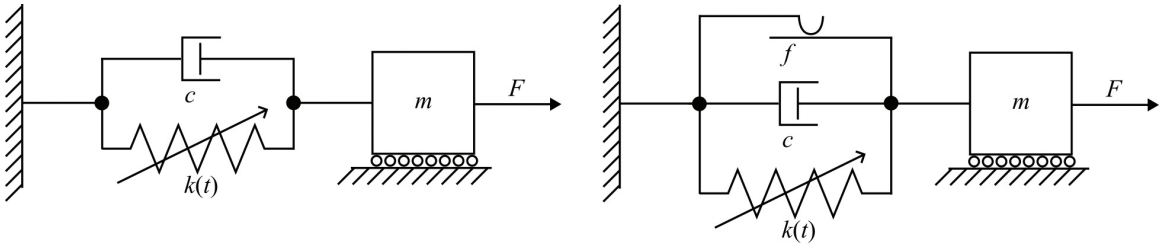


Fig. 4 Schematic of the case study system: (a) classical spring–mass–damper with time-varying stiffness $k(t)$ used as the assumed model for the system; (b) extended configuration with an additional nonlinear restoring force f used for developing the training dataset.

The input data for training are obtained by simulating multiple realizations of the system response under various forcing conditions and time-varying stiffness. Displacement, velocity, acceleration, and external force trajectories are collected at discrete time steps and organized following the preprocessing pipeline described in Algorithm 1. These trajectories form the training and validation datasets used to drive the learning process outlined in Algorithm 2. By embedding the physics directly through Eq. (1), the framework ensures that all predicted accelerations remain consistent with the governing dynamics, while the neural network focuses exclusively on estimating the hidden stiffness evolution.

Model performance is assessed by comparing predicted and measured accelerations through the mean squared error (MSE) of the data. Additionally, the reconstructed stiffness trajectories $\hat{k}(t)$ are benchmarked against ground truth values to evaluate how well the model captures the underlying parameter. The results presented in the following section demonstrate that the framework not only reproduces accurate dynamic responses but also yields interpretable stiffness estimates.

Simulated Stiffness Trajectories

Figure 5 presents the simulated stiffness evolutions used for training and validation across the 100 trajectories. As shown, stiffness values decrease nonlinearly. This diversity ensures that the training set captures a wide distribution of possible system behaviors. As highlighted in earlier sections, these stiffness histories are not directly available to the neural network during training but are used to benchmark the predictions.

Simulated Dynamic Response

Figure 6 shows how the response passes through resonance as stiffness decreases over time: the decay in stiffness (k) lowers the natural frequency (ω_{n}), bringing it into alignment with the frequency of the forcing function (F), producing the visible amplitude build-up near mid-simulation. As k continues to drop, ω_{n} moves past the forcing band and the oscillation amplitude subsides, consistent with the expected resonance-crossing behavior.

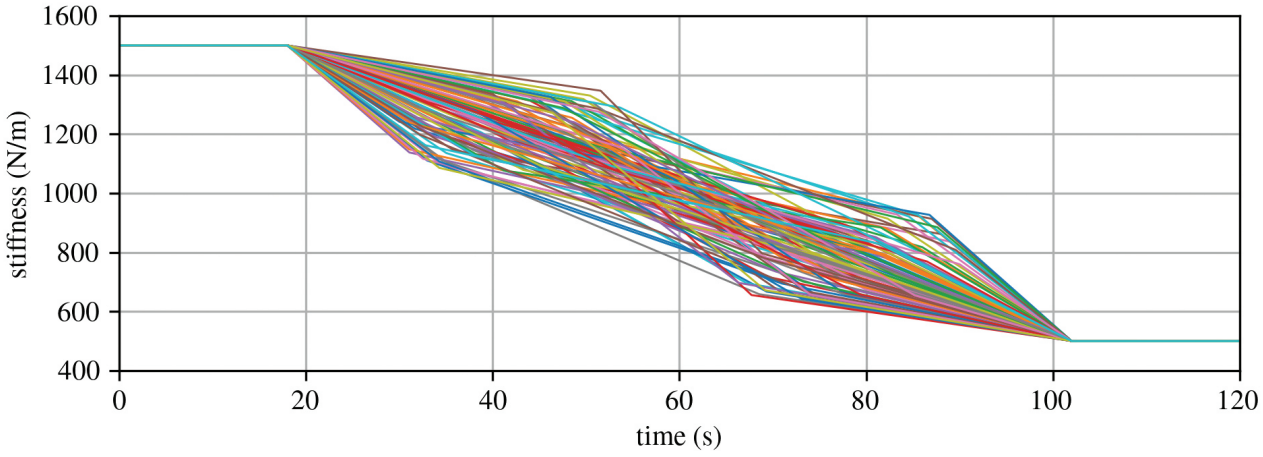


Fig. 5 True stiffness profiles $k(t)$ across 100 simulated trajectories of 120 s each. The variability in decay paths provide a diverse training and testing environment.

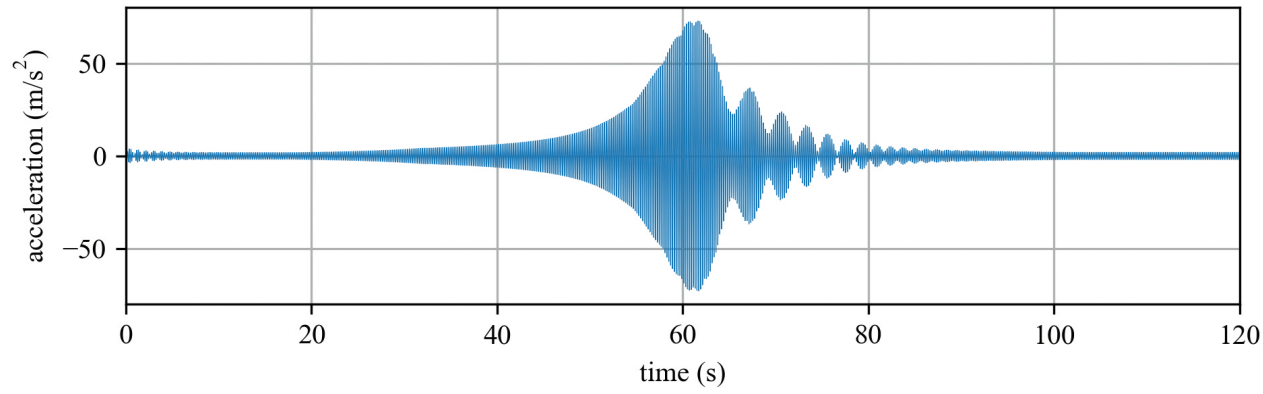


Fig. 6 Measured acceleration response for one test trajectory, showing how the system moves through resonance during a reduction in stiffness.

Inverse Stiffness Identification

Figure 7 compares three stiffness trajectories for a representative test case, the solely data-driven neural network, the physics-informed machine learning (PIML) that includes stiffness indirectly by minimizing the acceleration residual of the governing dynamics, and the physics-based ground truth. Both learning approaches reproduce the global trend of stiffness degradation and closely track the reference over the entire time span. The solely data-driven model yields a smooth estimate that aligns well with the ground truth, demonstrating strong interpolation capability within the training distribution. The physics-informed model, which embeds the spring, mass, and damper equations in the learning loop, sometimes exhibits fluctuations in the estimated stiffness. These arise from sensitivity to transient dynamics and reflect the model's insistence on satisfying the equations of motion at each step rather than fitting a smooth regression.

The close agreement between the PIML learned stiffness and the physics-based ground truth shown in Figure 7 suggests that the measured responses are well captured by the framework. While incorporating physics helps guide the solution space toward behavior consistent with the assumed dynamics, it does not guarantee correctness. The approach holds promise for extending inference beyond the systems and conditions represented in the training data, including under noise, parameter drift, or unseen loading. However, its success remains dependent on model adequacy, identifiability, and optimization stability.

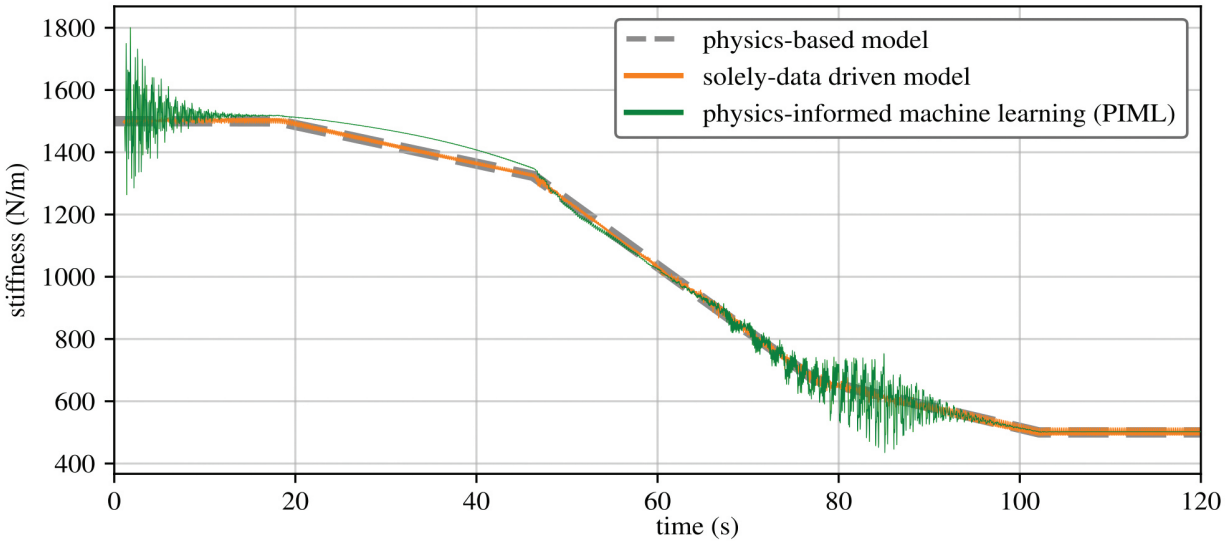


Fig. 7 Comparison of stiffness estimates for the considered test case.

Parameter Study

This section investigates the effect of hyperparameters over two different activation functions to predict the accuracy of the proposed PIML framework. Here, sigmoid and tanh are utilized as two widely used activation functions in the artificial neural network diagrammed in Figure 2. Table 2 reports the RMSE values for different network configurations where depths (number of hidden layers) and widths (neurons per layer) are the main tuning hyperparameters in evaluating the network performance.

For sigmoid, the linear-scale plot (Fig. 8a) highlights the steep error reduction when moving from shallow to deeper networks, while the logarithmic scale (Fig. 8b) makes it easier to distinguish fine differences within the low-error region. For tanh, the linear-scale surface (Fig. 8c) shows overall lower errors compared to sigmoid in shallow configurations, while the logarithmic view (Fig. 8d) emphasizes the robustness of deeper and wider models, where multiple settings converge to similar small RMSE values. By considering the values in Table 2, these plots demonstrate that while tanh offers advantages for shallow networks, both activation functions reach a stable low-error region in deeper and wider architectures.

In general, Figure 8 shows that depth beyond two layers yielded diminishing returns in our hyper-parameter search. Across both activations, error drops sharply when moving from one to two hidden layers, then flattens into a broad low-RMSE basin for 2–4 layers. For shallow networks, tanh achieves consistently lower errors than sigmoid. Increasing width (nodes per layer) helps, but with gentler, nearly monotonic improvements; most of the gain arrives by 50 nodes. The logarithmic views highlight a shallow gains along 2–4 layers and 50–100 nodes. The best accuracy is obtained with four layers and 100 neurons, though several nearby configurations perform nearly as well at lower computational cost.

Table 2 RMSE for sigmoid vs. tanh activation function across different network configuration

Neurons per layer	Number of hidden layers							
	Sigmoid activation				Tanh activation			
	1	2	3	4	1	2	3	4
10	6.20	1.20	2.00	0.90	3.36	0.68	0.37	0.35
25	3.60	0.18	0.30	0.19	2.42	0.14	0.09	0.08
50	1.42	0.13	0.09	0.05	1.83	0.06	0.05	0.04
75	1.23	0.07	0.05	0.04	1.56	0.08	0.04	0.05
100	1.07	0.06	0.04	0.04	1.40	0.05	0.05	0.04

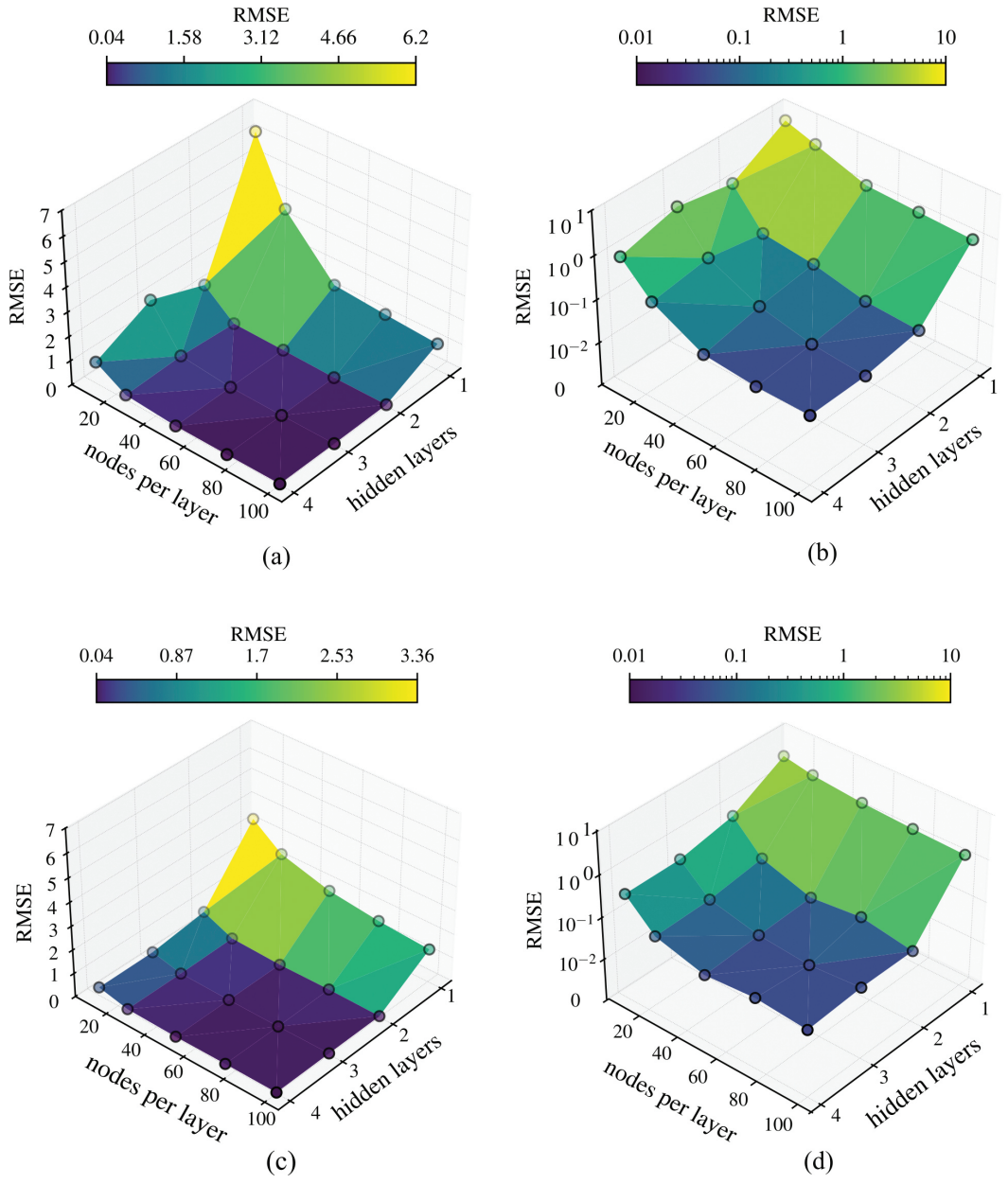


Fig. 8 RMSE surface as a function of hidden layers and neurons per layer for two activation functions. (a) Sigmoid with linear scale, highlighting the sharp error reduction. (b) Sigmoid with logarithmic scale, making small variations more visible. (c) Tanh with linear scale, showing generally lower RMSE than sigmoid in shallow networks. (d) Tanh with logarithmic scale, revealing stable low-error regions across deeper and wider architectures.

Conclusion

This paper presented a hard constraint physics-informed machine learning (PIML) framework for structural dynamics and demonstrated its effectiveness on a spring, mass, and damper system with time-varying stiffness. By embedding the governing equation of motion directly into the learning process, the proposed approach biases the solution space toward physically consistent behavior while retaining the flexibility and adaptability of data-driven models.The framework successfully recovers hidden stiffness trajectories that closely match the reference dynamics.

Overall, the hard-constraint PIML framework successfully recovered time-varying stiffness and reproduced key response features on the studied spring–mass–damper system. A simple hyperparameter sweep showed that modest depth (2–4 layers)

with moderate width is sufficient to drive test error to roughly $\mathrm{RMSE} \approx 0.04$, indicating practical accuracy without excessive model complexity.

There is a trade-off between the PIML model and a solely data-driven model. While a neural network trained solely on data can mimic stiffness trajectories with high accuracy, it lacks a true understanding of the underlying physics and may fail when exposed to conditions beyond its training distribution. In contrast, the PIML approach, though sometimes yielding responses that deviate from the reference trajectory, actively enforces the governing ODE constraints and therefore reflects a physically guided attempt to capture the system behavior. This enables PIML to generalize more reliably to unseen scenarios, operate effectively with less data, and produce predictions that are physically interpretable.

For the hard-constraint PIML considered here, there is clear promise for inverse problems: coupling measurements with the governing equation of motion to regularize otherwise ill-posed parameter estimates (e.g., time-varying stiffness). While promising, the main challenge observed in this study is the appearance of ripple artifacts in the inferred stiffness, arising from unsteady solutions when the equation of motion is enforced at every step within the learning loop.

Acknowledgments This material is based upon work supported by the National Science Foundation grant numbers CCF - 1956071, CCF-2234921, and CPS - 2237696. Additional support from the Air Force Office of Scientific Research (AFOSR) through award no. FA9550-21-1-0083. Any opinions, findings conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or the United States Air Force,

References

1. Fuh-Gwo Yuan, Sakib Ashraf Zargar, Qiuyi Chen, and Shaohan Wang. "Machine learning for structural health monitoring: challenges and opportunities". In Daniele Zonta, Hoon Sohn, and Haiying Huang, editors, *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2020*. SPIE, April 2020.
2. Elizabeth J. Cross, S. J. Gibson, M. R. Jones, D. J. Pitchforth, S. Zhang, and T. J. Rogers. *Physics-Informed Machine Learning for Structural Health Monitoring*, pages 347–367. Springer International Publishing, October 2021.
3. Marcus Haywood-Alexander, Wei Liu, Kiran Bacsa, Zhilu Lai, and Eleni Chatzi. "Discussing the spectrum of physics-enhanced machine learning: a survey on structural mechanics applications". *Data-Centric Engineering*, 5, 2024.
4. X. Liang, Y. Zhang, and J. Chen. "Physics-informed machine learning for structural dynamics: A review and outlook". *Mechanical Systems and Signal Processing*, 215:111246, 2024.
5. Daniel Coble, Liang Cao, Austin R.J. Downey, and James M. Ricles. "Physics-informed machine learning for dry friction and backlash modeling in structural control systems". *Mechanical Systems and Signal Processing*, 218:111522, September 2024.
6. Ying Zhou, Shiqiao Meng, Yujie Lou, and Qingzhao Kong. "Physics-informed deep learning-based real-time structural response prediction method". *Engineering*, October 2023.
7. Eleonora Maria Tronci, Austin R. J. Downey, Azin Mehrjoo, Puja Chowdhury, and Daniel Coble. *Physics-Informed Machine Learning Part I: Different Strategies to Incorporate Physics into Engineering Problems*, pages 1–6. River Publishers, August 2025.
8. Austin R. J. Downey, Eleonora Maria Tronci, Puja Chowdhury, and Daniel Coble. *Physics-Informed Machine Learning Part II: Applications in Structural Response Forecasting*, pages 63–66. River Publishers, August 2025.
9. M. Raissi, P. Perdikaris, and G.E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". *Journal of Computational Physics*, 378:686–707, 2019.
10. Eleonora Maria Tronci, Austin Downey, Conor Madden, Mohsen Gol Zardian, and Daniel Coble. *Physics-Informed Machine Learning Part IV: Weight Tuned Soft Constraint Method for Structural Dynamics*. August 2026.
11. Z. Lai, L. Lu, M. Dao, and G.E. Karniadakis. "Hard enforcement of interface and boundary conditions in physics-informed neural networks". *Computer Methods in Applied Mechanics and Engineering*, 404:115783, 2023.
12. R.T.Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud. "Neural ordinary differential equations". In *Advances in Neural Information Processing Systems*, pages 6571–6583, 2018.
13. L. Lu, P. Jin, G. Pang, Z. Zhang, and G.E. Karniadakis. "Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators". *arXiv preprint arXiv:1910.03193*, 2019.
14. Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. "Fourier neural operator for parametric partial differential equations". *arXiv preprint arXiv:2010.08895*, 2020.
15. Austin Downey and Laura Micheli. "Vibration mechanics: A practical introduction for mechanical, civil, and aerospace engineers". 2024.
16. ARTS-Lab. "Paper-2026-PIML-part-III-hard-constraint-ODE-method-for-structural-dynamics". GitHub, 2026. https://github.com/ARTS-Laboratory/Paper-2026-PIML-Part-III-Hard-Constraint-ODE-Method-for-Structural-Dynamics.

Chapter 7
Multi-Task Equation Discovery

S.C. Bee, K. Worden, N. Dervilis, and L.A. Bull

Abstract Equation discovery provides an approach to system identification by uncovering the governing dynamics of a system from observed data. By applying a Bayesian relevance vector machine (RVM) within a multi-task (MT) framework for simultaneous parameter identification across multiple datasets, this work seeks to improve the challenge of generalising models. In this work, responses of the same structure under different excitations are modelled as related tasks with shared parameters and task-specific noise. A simulated single degree-of-freedom oscillator with linear and cubic stiffness presents the case study, with datasets generated under three excitation regimes. This work shows that while single-task RVMs reproduce responses, they often miss governing terms when nonlinear dynamics were weakly excited. In contrast, the MT-RVM leverage information across tasks, improving parameter recovery under weak and moderate excitations while retaining accuracy at high excitation. These results show how multi-task Bayesian inference can reduce over-fitting and improve generalisation in equation discovery, with particular relevance to structural health monitoring under variable loading.

Keywords Sparse regression · Equation discovery · Multi-task learning · System identification

Introduction

Equation discovery has been applied to the field of system identification to identify equations of motion that capture the behaviour of measured systems, along with the parameters associated with each of their terms [1]. In practice, however, any selected model is only an approximation of reality, and uncertainty is inevitably present in the predictive equations [2]. To address this, Bayesian approaches have been widely explored for equation discovery, which quantify parameter uncertainty [2–6].

A potential challenge in equation discovery is the risk of over-fitting, where the discovered model fits the training data closely but fails to represent the underlying physical system. The antithesis of over-fitting is generalisation, which can be encouraged via several strategies. One way to aid generalisation is via *multi-task learning* (MTL), which learns multiple tasks simultaneously and exploits shared information to capture underlying commonalities. Caruana outlines several domains of MTL [7], some of which have since been translated into the context of structural health monitoring (SHM) [8].

In this paper, MTL is applied by treating datasets from the same structure under different excitation levels as separate, but related, tasks. By sharing parameters across tasks, the model exploits common features while still accounting for differences introduced by varying excitations. This design ensures that the model does not over-fit to one excitation condition (one dataset) and instead learns a representation of the system that is valid across multiple responses.

This approach is valuable in SHM, where different loading conditions often highlight different aspects of the same underlying physics. For example, low-level excitations may reveal the system's linear response, while higher excitations expose nonlinear effects. By training across multiple excitation scenarios simultaneously, the model is able to combine these complementary insights into a single, coherent description of the structure.

S.C. Bee · K. Worden · N. Dervilis
Dynamics Research Group, School of Mechanical, Aerospace and Civil Engineering,
University of Sheffield, Mappin Street, Sheffield, S1 3JD, UK
e-mail: scbee1@sheffield.ac.uk; k.worden@sheffield.ac.uk; n.dervilis@sheffield.ac.uk

L.A. Bull
School of Mathematics and Statistics, University of Glasgow, Glasgow, G12 8QQ, UK
e-mail: Lawrence.Bull@glasgow.ac.uk

© The Author(s), under exclusive license to River Publishers 2026

Kristin B. Zimmerman, PhD (eds.), *Data Science in Engineering, 2026, Vol. 9*, Conference Proceedings
of the Society for Experimental Mechanics Series, https://doi.org/10.13052/rp-9788743814207

System Identification with Bayesian Inference

System identification characterises the dynamics of a system from observations of the evolution of states through time. The current work follows the approach proposed by [1] and treats equation discovery as sparse linear regression, known as SINDy.

For a multi-degree-of-freedom system, the system states are typically displacement and velocity,

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y \\ \dot{y} \end{bmatrix} \tag{1}$$

where y is displacement, $\dot{y}$ is velocity and $\dot{}$ denotes the first order differentiation with respect to time.

Within the framework of equation discovery, the above formulation is combined with a predefined dictionary, or design matrix, of M candidate terms,

$$\mathbf{D} = [d_1(\boldsymbol{X}), d_2(\boldsymbol{X}), ..., d_M(\boldsymbol{X})] \tag{2}$$

where $\boldsymbol{X} = [\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_N}]$ and $\mathbf{D} \in R^{N \times M}$. This corresponds to N data-points and M candidate components (features). The design matrix is intentionally over-defined, incorporating a wide range of potential components that could contribute to the equation of motion of the system.

By utilising shrinkage [9], the algorithm identifies only the base elements that are most relevant, effectively revealing the governing dynamics of the system.

RVM Formulation of SINDy

The relevance vector machine (RVM), devised by Tipping [9], is a Bayesian algorithm which has a similar functional form to the deterministic support vector machine [10]. RVMs have been used within various applications to analyse systems, and the approach is well-suited to equation discovery problems. The formulation used in this research is demonstrated graphically in Figure 1.

In the context of identifying forced (non-autonomous) duynamical systems, a function $\mathbf{F}$ is present, which represents the input force acting on the structure. Arguably, this could be embedded within the design matrix, defined in equation (2); however, in this work it is treated as a separate observed variable,

$$\boldsymbol{y} = \mathbf{D}\mathbf{w} + \mathbf{F} + \boldsymbol{\epsilon} \tag{3}$$

where $\mathbf{w} \in R^M$ is the vector representing the weights associated with each candidate basis function and $\boldsymbol{\epsilon}$ is the noise that distorts the signal.

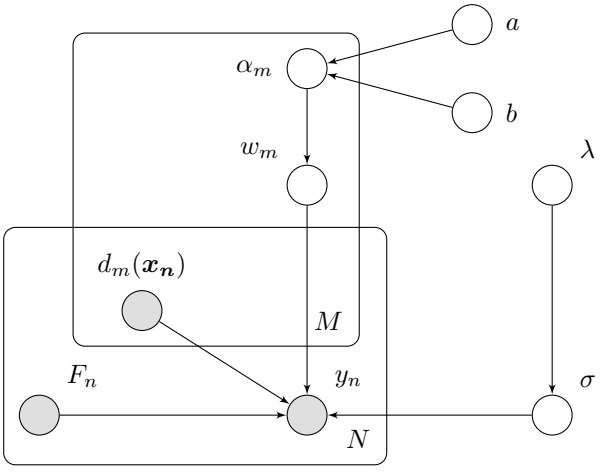


Fig. 1 A graphical representation of the RVM. Grey nodes correspond to observed variables and white nodes represent latent variables. The M plate represents the number of candidate features and the N plate represents the number of data points

Each weight is assumed to follow a zero-mean Gaussian prior, $w_m \sim \mathcal{N}(0, \alpha)$ where the variance is governed by an inverse gamma distribution, $\alpha_m^2 \sim \Gamma^{-1}(a, b)$.

This hierarchical prior structure introduces hyper-parameters a and b, which control the distribution of weight variances and encourage sparsity.

The signal noise is assumed to be Gaussian, $\epsilon \sim \mathcal{N}(0, \sigma^2)$, with an exponential prior placed on the variance of the noise, $\sigma^2 \sim \text{Exp}(\lambda)$, where λ is a hyper-parameter. This prior has a preference for data with generally low levels of noise.

RVM Extension to Multiple Tasks

When analysing L tasks, the RVM can be adapted as shown in Figure 2.

Figure 2 shows that the weight vector, $\mathbf{w}$, is shared between *all* tasks, while the noise variance is task-specific. A shared weight distribution reflects the assumption that measurements across tasks have common underlying physical properties. In contrast, the noise is modelled separately for each task as it typically arises from environmental conditions or measurement settings that vary between tasks.

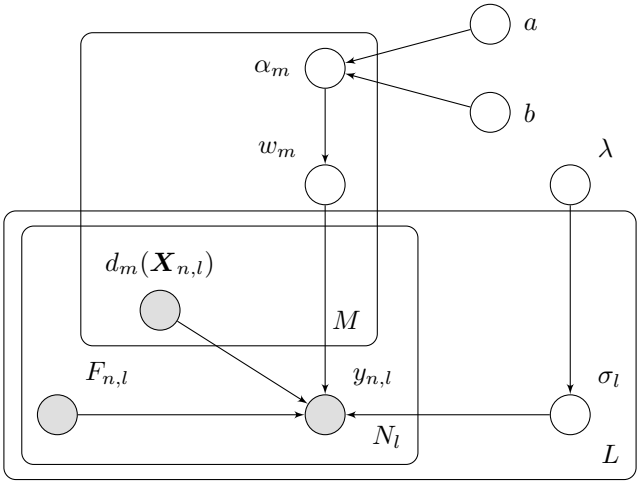


Fig. 2 A graphical representation of a multi-task RVM. Grey nodes correspond to observed variables and white nodes represent latent variables. The L plate represents the number of tasks, the M plate represents the number of candidate features and the N plate represents the number of data points

Case Study

A simulated dataset was developed, based on a single degree-of-freedom (SDOF) system with nonlinear stiffness, shown by Figure 3.

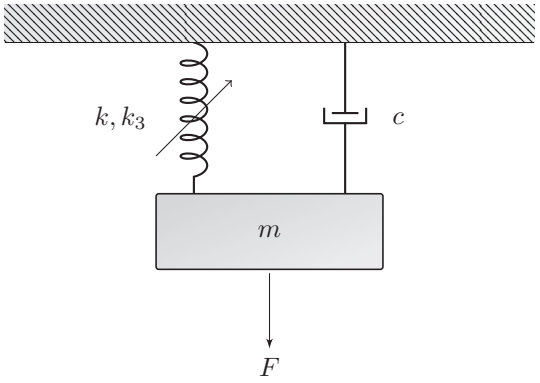


Fig. 3 SDOF oscillator with stiffness (linear, k and cubic, k_3), mass m, damping coefficient c and force acting on the system, F

The equation of motion for this system is given by,

$$m\ddot{y} + c\dot{y} + k_1 y + k_3 y^3 = F \tag{4}$$

where $m = 1kg$ is mass, $c = 0.2Ns/m$ is the damping coefficient, $k_1 = 1N/m$ is the linear stiffness, $k_3 = 1N/m^3$ is the cubic stiffness, F is the system forcing, and, $\ddot{}$ denotes the second order differentiation with respect to time. The values for the damping coefficient and stiffnesses were selected to demonstrate the algorithm and not to be representative of a real system.

The system was integrated forward in time via a fourth-order Runge-Kutta method, with a sampling frequency of 100kHz. To reduce dataset size, the signal was downsampled by selecting every thousandth point, resulting in an effective sampling frequency of 100Hz over an interval of 10s. A Butterworth filter [11] was applied with a low cut-off of 3Hz.

To generate a suitable dataset, it was necessary to construct data under varying conditions. Since the objective is to use these datasets to gain insight into the structure, the underlying physics, represented by the right-hand side of equation (4), must remain unchanged. Consequently, only the input forcing can be modified to produce different datasets. The forcing is modelled as random excitation in the form of noise, drawn from a uniform distribution whose amplitude can be scaled by a factor, f, such that $F \sim f\mathcal{U}(-0.5, 0.5)$. To create three distinct datasets, data were generated over a duration of 10s with $f = \{10^1, 10^2, 10^3\}$. Three datasets allow for a single task (ST) scenario with low, medium and high excitation, one at each of the values of f, and one multi-task (MT) scenario with all three datasets utilised in a joint inference for the identification of the shared underlying system.

For each of the three datasets, the dictionary of candidate basis functions required for the RVM was defined as,

$$D(\mathbf{X}) = \begin{bmatrix} | & | & | & | & | & | & | \\ \mathbf{y} & \dot{\mathbf{y}} & \mathbf{y}^2 & \dot{\mathbf{y}}^2 & \mathbf{y}^3 & \dot{\mathbf{y}}^3 & \mathbf{1} \\ | & | & | & | & | & | & | \end{bmatrix} \tag{5}$$

Results and Discussion

It is worth noting that the target vector, $\boldsymbol{y}$ (equation (3)), for this problem is acceleration, $\ddot{y}$. Hence, Equation (4) will take the form,

$$m\ddot{y} = F - c\dot{y} - k_1 y - k_3 y^3 \tag{6}$$

The system has unit mass, and therefore, the target value of $\boldsymbol{w}$ will be: $[-k_1, \quad -c, \quad 0, \quad 0, \quad -k_3, \quad 0, \quad 0]$.

Figure 4 shows the results for $f = 10^1$. The lower plot shows that the RVM output tracks the target acceleration over the 10-second training dataset reasonably well. However, the estimated values of the elements in $\boldsymbol{w}$, shown in the upper plot, deviate from the true values. The mean of the displacement cubic term, y^3, is 0.79, much closer to unity than the true value of minus unity and the mean of the velocity cubic term, $\dot{y}^3$, is -0.77, closer to -1 than the true value of 0, this outcome is challenging to interpret in physical terms.

Figure 5 shows the results for $f = 10^2$ and Figure 6 shows the results for $f = 10^3$. As with the lower plot in Figure 4, the corresponding plots in both Figure 5 and Figure 6 accurately tracks the target acceleration over the 10-second training dataset.

However, for $f = 10^2$ the estimated weight matrix again diverges from the *True* values. In this case, y^3, is 0.01, much closer to zero than the true value of -1, this discrepancy can be attributed to the insufficient excitation of the system's nonlinearities. The model misidentified several terms because of insufficient excitation, yielding values that deviate from the *True* dynamics and lack clear physical interpretation.

The weight matrix for $f = 10^3$ is in good agreement with the *True* values of $\boldsymbol{W}$, the value for y^3 is slightly lower than the *True* value but is much closer than when $f = 10^2$ and $f = 10^3$.

The results for the combination of all the data as three distinct tasks is shown in Figure 7. The prediction of values over the 10s interval for the three tasks remains good and there is improvement in the prediction of W for $f = 10^1$ and $f = 10^2$. The results of $f = 10^3$ and the multi-task learner are similar indicating that there has been positive knowledge transfer from $f = 10^3$ to the other two tasks, without negative transfer from the lower excitation tasks impacting the strongest performing task.

The parameter used to evaluate the predictive performance of each model is the normalised mean-square error metric,

$$NMSE = \frac{100}{N\sigma_z^2}\Sigma_{n=1}^{N}(z_n - \hat{z}_n)^2 \tag{7}$$

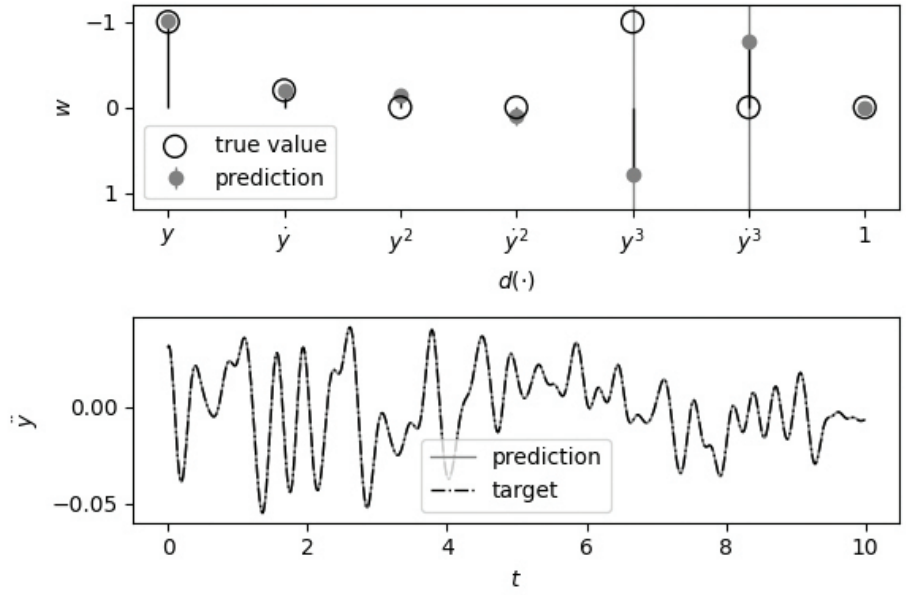


Fig. 4 Results of the ST scenario with medium excitation, $f = 10^2$ with a standard RVM.(*Upper*) Weight matrix, $\boldsymbol{W}$, results; (*Lower*) the predicted response

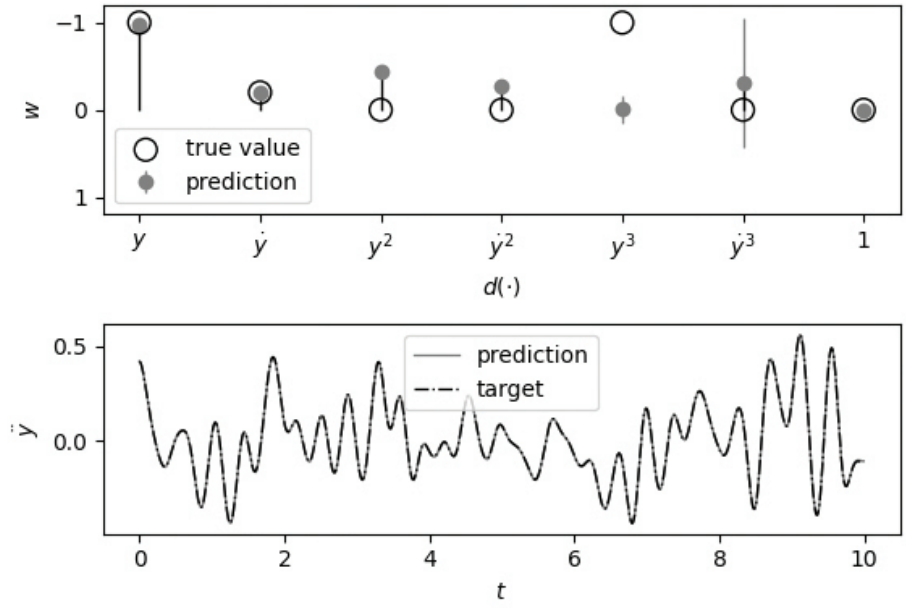


Fig. 5 Results of the ST scenario with medium excitation, $f = 10^1$ with a standard RVM as per Figure 1).(*Upper*) Weight matrix, $\boldsymbol{W}$, results; (*Lower*) the predicted response

σ_z^2 is the variance of the target, $\mathbf{z}$, $\hat{\cdot}$ represents the predicted value, and, in this study $\mathbf{z} = \ddot{\boldsymbol{y}}$.

Figure 8 shows the NMSE results from the ST and MT scenarios for the three excitation levels. One hundred datasets were generated at each excitation level and the NMSE for each dataset was calculated. The graph shows the average and distribution of the NMSE for the different excitation levels and the different models.

For the low and high excitation cases, the NMSE is similar between the ST and MT models. The benefit of the MT model is demonstrated in the medium excitation case as the average NMSE is lower, but also, the variance is lower. With medium excitation, the data does not contain information on the nonlinear components of the structure, as these nonlinear components have not been excited. Therefore, the ST model with medium excitation has been unable to identify this nonlinearity and

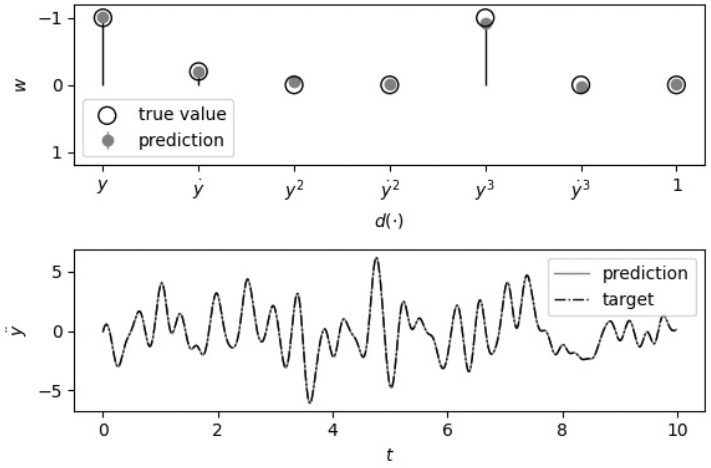


Fig. 6 Results of the ST scenario with high excitation, $f = 10^3$ with a standard RVM.(*Upper*) Weight matrix, $\boldsymbol{W}$, results; (*Lower*) the predicted response

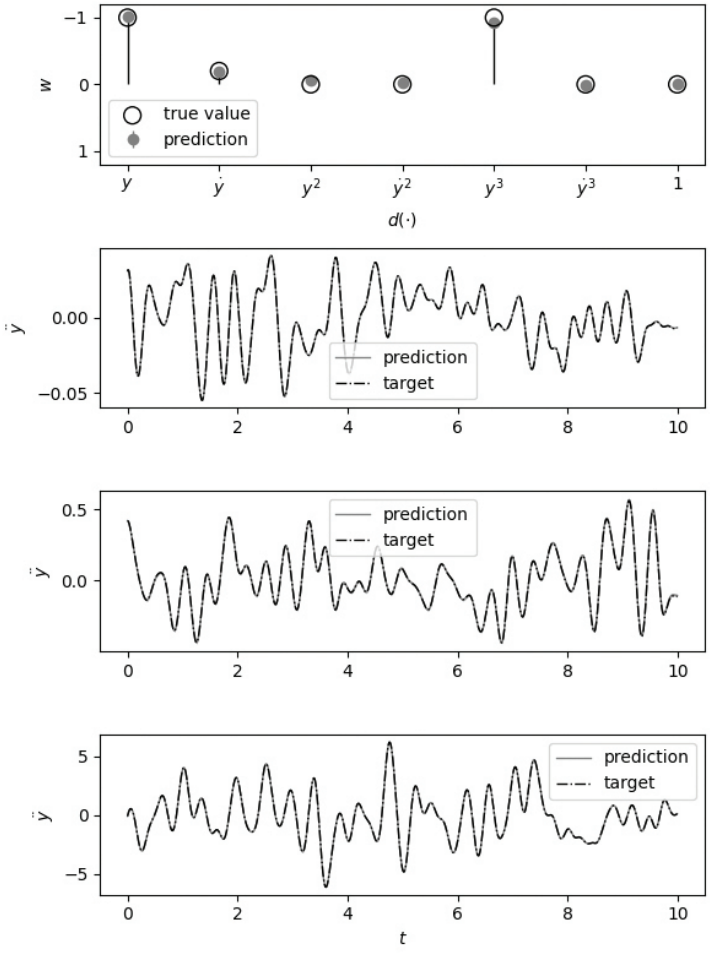


Fig. 7 Results of the MT scenario with $f = [10^1 \quad 10^2 \quad 10^3]$ with a multi-task RVM (as per Figure 2); (*Upper*) Weight matrix, $\boldsymbol{W}$, results shared across all tasks; (*Second*) the predicted response for $f = 10^1$; (*Third*) the predicted response for $f = 10^2$; and, (*Fourth*) the predicted response for $f = 10^3$

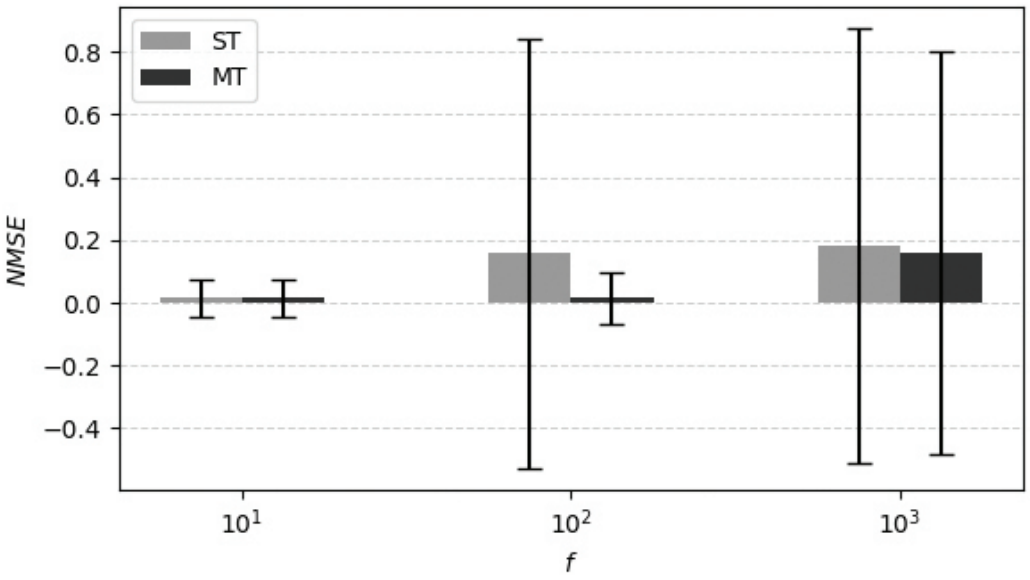


Fig. 8 The distribution of NMSE for the MT and ST scenarios showing 3σ

instead, has fit a model based on structurally incomplete data (missing nonlinear information). Hence, the ST model's weight matrix does not match well with actual parameters. In contrast, the MT model *does* have a dataset which includes both linear and nonlinear excitation, the parameters are much closer to the *True* values.

Conclusion

This provides a strong example of how multi-task learning can enhance equation discovery by promoting generalisation across tasks. When working with a single task, it is often difficult to assess whether the identified model truly generalises. For instance, the solutions obtained for low levels of forcing, $f = 10^1$ and $f = 10^2$, differ considerably between the two single-task models. By contrast, the multi-task RVM yields a solution that performs well across all three datasets. This consistency suggests that the model has captured a representation that better reflects the underlying physics of the system. When considering the NMSE of the models, the MT model outperforms single-task models with a lower standard deviation in the medium and high excitation cases.

Acknowledgments The authors wish to gratefully acknowledge support for this work through grants from the Engineering and Physical Sciences Research Council (EPSRC), UK, and Natural Environment Research Council, UK via grant number, EP/S023763/1. For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

References

1. S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.
2. J. L. Beck, "Bayesian system identification based on probability logic," *Structural Control and Health Monitoring*, vol. 17, no. 7, pp. 825–847, 2010.
3. K. Worden and J. Hensman, "Parameter estimation and model selection for a class of hysteretic systems using Bayesian inference," *Mechanical Systems and Signal Processing*, vol. 32, pp. 153–169, 2012. Uncertainties in Structural Dynamics.
4. R. Fuentes, N. Dervilis, K. Worden, and E. Cross, "Efficient parameter identification and model selection in nonlinear dynamical systems via sparse Bayesian learning," *Journal of Physics: Conference Series*, vol. 1264, p. 012050, jul 2019.
5. R. Fuentes, R. Nayek, P. Gardner, N. Dervilis, T. Rogers, K. Worden, and E. Cross, "Equation discovery for nonlinear dynamical systems: A Bayesian viewpoint," *Mechanical Systems and Signal Processing*, vol. 154, p. 107528, 2021.
6. Y.-C. Zhu, P. Gardner, D. J. Wagg, R. J. Barthorpe, E. J. Cross, and R. Fuentes, "Robust equation discovery considering model discrepancy: A sparse Bayesian and Gaussian process approach," *Mechanical Systems and Signal Processing*, vol. 168, p. 108717, 2022.

7. R. Caruana, “Multitask learning,” *Machine Learning*, vol. 28, pp. 41–75, 07 1997. Copyright - Kluwer Academic Publishers 1997; Last updated - 2023-11-29.
8. S. Bee, L. Bull, N. Dervilis, and K. Worden, “When is an SHM problem a multi-task-learning problem?,” 2023.
9. M. E. Tipping, “Sparse Bayesian learning and the relevance vector machine,” *J. Mach. Learn. Res.*, vol. 1, p. 211–244, Sept. 2001.
10. D. Kong, Y. Chen, N. Li, C. Duan, L. Lu, and D. Chen, “Relevance vector machine for tool wear prediction,” *Mechanical Systems and Signal Processing*, vol. 127, pp. 573–594, 2019.
11. S. Butterworth, “On the theory of filter amplifiers,” *Experimental Wireless & the Wireless Engineer*, vol. 7, pp. 536–541, Oct. 1930.

Chapter 8
Physics-Informed Machine Learning Part IV: Weight-Tuned Soft-Constraint Method for Structural Dynamics

Eleonora Maria Tronci, Austin R. J. Downey, Connor Madden, Mohsen Gol Zardian, and Daniel Coble

Abstract Physics-informed machine learning (PIML) is a methodology that combines principles from physics with machine learning (ML) techniques to enhance the accuracy and interpretability of predictive models. By incorporating physical laws and constraints into the learning process, physics-informed machine learning enables more robust predictions and reduces the need for large amounts of training data. PIML has a wide range of applications in science and engineering, such as modeling physical systems, solving partial differential equations, and performing inverse analysis and optimization.

In Part IV of this series, the authors describe a soft-constraint, physics-informed neural network (PINN) tutorial for structural dynamics. Instead of enforcing equilibrium exactly at every step, the network is trained with a composite loss that combines several data fitting terms (displacement, velocity, acceleration, and stiffness taken from an experimental window) with a single physics consistency term that measures how closely the predicted forces satisfy the mass–damping–stiffness relationship. Each term carries an explicit weight, allowing the decision on how much trust to place in measured data versus the governing equation. A compact multilayer perceptron takes as input the displacement, velocity, and external force from the most recent timestep and predicts the next-step displacement along with an effective stiffness. Two weight tuning options are covered: (i) a manual recipe that highlights how shifting emphasis from data to physics improves extrapolation but can amplify noise, and (ii) a self-adaptive schedule that raises the physics weight whenever the residual grows and relaxes it once the residual falls below a user-selected tolerance, eliminating the need for grid searches.

The technique is demonstrated using the same spring–mass–damper system introduced in Part III, but the PINN is trained on a single experiment featuring a specific time-varying stiffness path. Because the model includes explicit time as an input, it becomes specific to that trajectory. Performance is benchmarked against two reference solvers: a data-only neural network trained without any physics guidance and the hard constraint ODE layer introduced previously. The paper concludes with qualitative guidelines for selecting initial weight ratios, determining the residual tolerance for the adaptive schedule, and monitoring convergence, providing readers with a practical roadmap for balancing data fidelity and physical plausibility in their own inverse problem applications.

Keywords Physics-informed machine learning · Physics-informed neural network · Soft-constraint · Weight tuning · Adaptive tuning

Eleonora Maria Tronci
Department of Civil and Urban Engineering, New York University, New York, NY, USA
e-mail: eleonora.m.tronci@gmail.com

Eleonora Maria Tronci
Center for Urban Science and Progress, New York University, New York, NY, USA
e-mail: eleonora.m.tronci@gmail.com

Austin R. J. Downey · Connor Madden · Mohsen Gol Zardian
Department of Mechanical Engineering, University of South Carolina, Columbia, SC, USA
e-mail: AUSTINDOWNEY@sc.edu; CMMADDEN@email.sc.edu; GOLZARDM@email.sc.edu

Austin R. J. Downey
Department of Civil and Environmental Engineering, University of South Carolina, Columbia, SC, USA
e-mail: AUSTINDOWNEY@sc.edu

Daniel Coble
Department of Mechanical Engineering & Materials Science, Duke University, Durham, NC, USA
e-mail: daniel.coble@duke.edu

© The Author(s), under exclusive license to River Publishers 2026

Kristin B. Zimmerman, PhD (eds.), *Data Science in Engineering, 2026, Vol. 9*, Conference Proceedings of the Society for Experimental Mechanics Series, https://doi.org/10.13052/rp-9788743814207

Introduction

Scientific modeling has traditionally been divided between two distinct paradigms [1, 2]. Physics-based numerical methods discretize governing equations to provide high-fidelity solutions, but are often constrained by prohibitive computational costs, complex mesh generation, and an inability to account for model-form uncertainty in the physical laws themselves. Conversely, data-driven models excel at learning complex patterns from data but typically lack physical interpretability, require massive training datasets, and struggle to generalize reliably, often producing physically implausible predictions outside of their training distribution.

Physics-informed machine learning (PIML) has emerged as a powerful paradigm for developing predictive models of physical systems, offering a compelling complement to purely data-driven or traditional physics-based approaches [3, 4]. By integrating governing physical laws directly into the learning pipeline, PIML provides a guided strategy to steer solutions toward physical behavior under sparse, noisy, or incomplete data. PIML is not inherently more accurate than conventional machine learning; its impact is highly context-dependent. Where abundant, representative data are available and the task remains within the training distribution, a well-regularized data-only model may match or exceed PIML in standard error metrics. Where interpretability, parameter recovery, or cautious extrapolation are required, the physics guidance can confer advantages by constraining the hypothesis space toward feasible dynamics. At the same time, recent analyses have shown that PINNs can exhibit imbalanced loss dynamics and stiffness during training, motivating principled strategies for calibrating the relative influence of data and physics terms [5]. This methodology is particularly effective for both forward problems, such as solving differential equations, and inverse problems, such as system identification and parameter estimation.

The domain of PIML is extensive and can be categorized in multiple ways [6]. However, physical principles can be incorporated in two primary ways: through *hard* or *soft* constraints. Hard-constraint approaches, as detailed in Part III of this series [7], embed the governing equations directly into the model's architecture (e.g., via a differentiable ODE integrator), encouraging the predicted solution to satisfy physical laws at each step. This strong enforcement has been shown to be successful in many applications [8], can improve stability and identifiability, but it may be less flexible under model-form uncertainty and can be more challenging to implement for systems with unmodeled dynamics or discontinuities. In contrast, soft-constraint approaches, presented here in Part IV, offer a more flexible and widely adopted alternative: the residual of the governing equations is incorporated as a penalty term in the model's loss function. Variants such as augmented-Lagrangian PINNs further reduce sensitivity to hand-tuned penalties by learning multipliers that tighten constraint satisfaction [9]. The choice between hard and soft constraints is problem-specific and depends on factors such as the fidelity of the governing model, noise characteristics, label availability for latent parameters, and the degree of distribution shift anticipated at inference.

Structural dynamics presents a particularly compelling application area for several PIML approaches [10]. Systems such as bridges, buildings, wind turbines, and aerospace structures obey well-established principles but often exhibit time variability, environmental influences, and mild nonlinearities that complicate pure first-principles modeling. In this context, soft-constraint physics-informed neural networks (PINNs) are attractive because they preserve a simple network architecture while injecting domain knowledge through a residual penalty on the equation of motion. Concrete structural-dynamics uses of residual-penalized (soft) physics terms include damage identification and system identification. For damage identification, Wang et al. [11] introduce a physics-guided residual network that augments a data-driven backbone with an equation-of-motion residual term, improving robustness under scarce/noisy measurements—an archetypal soft-constraint strategy for SHM tasks. For inverse structural dynamics, Liu et al. formulate a PINN that recovers parameters in systems with multi-physics damping (nonlinear dissipation) by minimizing a composite objective that mixes data misfit with residual penalties, illustrating soft-constraint PINNs for parameter estimation in realistic structural models [12]. In forward vibration analysis, Chen et al. develop a time-marching PINN tailored to long-duration simulations of structural vibrations. Although focused on efficiency, their formulation is still driven by residual penalties rather than hard architectural enforcement, underscoring the practicality of soft-constraints in vibration problems [13].

The successful implementation of soft-constraint depends on balancing multiple loss components, such as data fidelity (e.g., displacement, velocity, acceleration, stiffness) versus physics consistency (equation-of-motion residual). These weights govern the effective bias–variance trade-off, influence the identifiability of latent parameters, and shape the optimization dynamics. Adaptive weighting schemes offer principled methods for adjusting this balance during training: Self-Adaptive PINNs learn per-sample physics weights that emphasize stubborn residuals [14]; dynamic weight strategies and self-adaptive loss balancing adjust term contributions online [15, 16]; and multi-task formulations provide global, uncertainty-based weights or gradient-normalization baselines that reduce manual tuning [17, 18]. These developments directly address the training imbalances highlighted in recent analyses [5] and motivate the weighting strategies examined in this work.

Recent contributions by the authors have examined multiple strategies for integrating physical knowledge into machine learning. Part I of this series outlined how embedding governing equations can enhance robustness, improve interpretability, and in many cases strengthen predictive performance in engineering settings [6]. Part II demonstrated a practical application

to structural dynamics, showing that physics-informed architectures can effectively forecast system responses [19]. Together, these studies highlight the benefits of coupling physical models with neural networks for inverse problems, including the estimation of system parameters.

This paper, Part IV of this series, presents a practical tutorial on applying a soft-constraint PINN to an inverse problem in structural dynamics. We use the same spring-mass-damper system with time-varying stiffness introduced in Part III [7] to provide a direct comparison of methodologies. The core of our contribution is a detailed exploration of the composite loss function, which combines data-fitting terms with a physics-consistency term derived from the equation of motion. We investigate two distinct weight-tuning strategies: (i) a manual recipe that highlights the fundamental trade-off between data fidelity and physical regularization, and (ii) a self-adaptive schedule that automates the balancing process. The performance of the soft-constraint PINN is benchmarked against a data-only neural network and the hard-constraint ODE model from Part III. We conclude with practical guidelines for selecting initial weight ratios and monitoring convergence, providing a clear roadmap for applying this flexible PIML technique to other inverse problems.

Methodology

This work develops a soft-constraint PIML framework to identify and predict the dynamic response of a mass–spring–damper system. The central idea is to combine a data-driven neural network with a physics-consistency term in the loss such that the governing equation of motion influences learning without being enforced exactly at every time step. Unlike hard-constraint approaches, where the ODE is encoded directly into the model architecture and satisfied identically, the present framework penalizes the residual of the governing composite loss function during training. Concretely, the model minimizes a composite objective that blends data-fitting terms (e.g., displacement, velocity, acceleration, and stiffness from an experimental window) with a physics residual penalty evaluated on the network's predictions. The relative contribution of each term is controlled by explicit weights, enabling practitioners to calibrate the trade-off between data fidelity and physical plausibility. As a result, the learned solution is not guaranteed to satisfy the ODE exactly but is regularized toward physically consistent behavior, and training is guided by both data and physics.

Data preparation

We adopt the same variable conventions as Part III: time t, displacement x_t, velocity v_t, acceleration a_t, stiffness k_t, and external force F_t are recorded as synchronized columns in tabular files (rows = time steps, columns = physical quantities). Raw trajectories from multiple experiments are stacked into a three-dimensional array $X \in \mathbb{R}^{N\times L\times D}$, where N is the number of trajectories, L the samples per trajectory, and D the number of measured channels. For continuity and visual reference, we reproduce the same schematic as Part III in Fig. 1 and defer full pseudocode and hyperparameter details (e.g., window length choices, batching policies) to Part III [7].

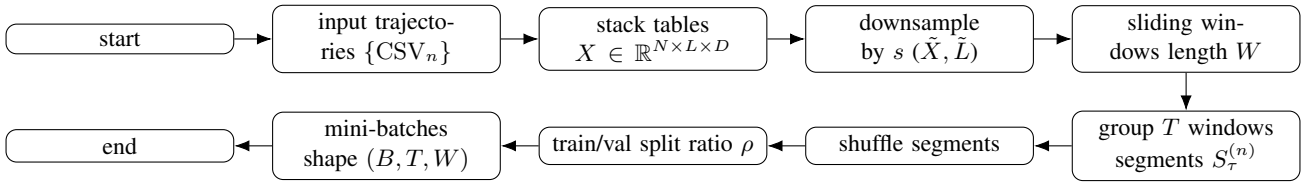


Fig. 1 Preprocessing pipeline from raw trajectories to mini-batches (verbatim labels from Part III)

Consistent with our implementation, we downsample all signals to a uniform rate (50 Hz) to reduce redundancy while preserving the dynamics of interest; let $\tilde{X}$ denote the downsampled tensor and $\tilde{L}$ the corresponding length. Basic sanity checks (monotone timestamps, consistent units) are applied prior to batching.

The main deviation from Part III concerns the supervision format. Part III forms fixed-length sliding windows of size W and groups T consecutive windows into segments to train the hard-constraint ODE layer. In contrast, Part IV (soft-constraint PINN) uses step-ahead supervision: for each time index t we form input–target pairs

$$\underbrace{[\,x_t,\ v_t,\ F_t,\ t\,]}_{\text{inputs}} \longrightarrow \underbrace{[\,x_{t+1},\ k_t\,]}_{\text{targets}},$$

optionally including a_t if available. This pairing directly supports the one-step predictor used in our model, while leaving the rest of the pipeline unchanged.

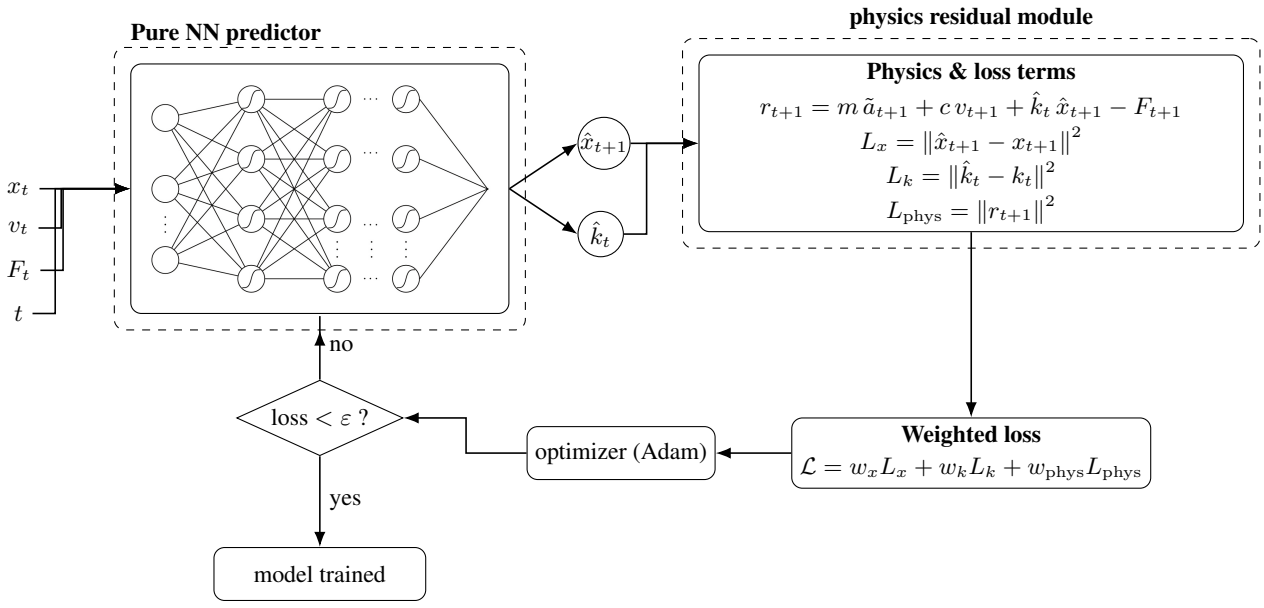


Fig. 2 Soft-constraint PINN: the NN predicts $(\hat{x}_{t+1}, \hat{k}_t)$. The physics module computes the residual and lists $L_x, L_k, L_{\text{phys}}$; the composite loss sits directly below it, the optimizer below that, and the stopping rule is placed to the left of the composite loss

To avoid temporal leakage within an experiment, we split $\tilde{X}$ by experiments (80/20 train/validation), then shuffle the training set and construct mini-batches. Batches thus have shape (B,D) for inputs and (B,D_{target}) for targets under the step-ahead pairing, in place of the (B,T,W) layout used in Part III's windowed segments. Apart from this pairing change, the stack $\rightarrow$ downsample $\rightarrow$ split $\rightarrow$ batch flow remains identical to Part III.

Network architecture and physics-embedded loss function

The proposed architecture follows a soft-constraint PIML formulation in which the governing law contributes through a residual penalty in the objective function. The present model employs a compact data-driven predictor together with a physics-consistency term evaluated at the same time index.

While PINNs are most often used for interpolation/forward problems, they can also be adapted to inverse settings when the physics is sufficiently known. Here we train on a test without friction so that the governing equation is fully specified (mass–damping–stiffness), and the soft-constraint acts as a physics-guided regularizer for estimating a time-varying stiffness. As is typical for forward-type PINNs, each trained model is specific to a single experiment/trajectory (i.e., not expected to generalize across materially different tests without retraining).

Let $z_t = [\,x_t,\, v_t,\, F_t,\, t\,]$ denote the input at time t (acceleration a_t may be included when available). A multilayer perceptron $\mathcal{N}_\theta$ advances the state by one time step and simultaneously estimates an effective stiffness,

$$\left[\,\hat{x}_{t+1},\ \hat{k}_t\,\right] \;=\; \mathcal{N}_\theta\big(z_t\big) \;\equiv\; \phi\big(t, x_t, v_t, F_t\big),$$

so that $\phi_1 = \hat{x}_{t+1}$ and $\phi_2 = \hat{k}_t$. When derivative consistency terms are used, autodifferentiation provides the required partial derivatives of ϕ_1 with respect to its inputs; time derivatives then follow by the chain rule, e.g.

$$\frac{d\phi_1}{dt} = \frac{\partial\phi_1}{\partial t} + \frac{\partial\phi_1}{\partial x}\,\dot{x} + \frac{\partial\phi_1}{\partial v}\,\dot{v} + \frac{\partial\phi_1}{\partial F}\,\dot{F}, \quad \frac{d^2\phi_1}{dt^2} \text{ analogously,}$$

with $\dot{x}, \ddot{x}, \dot{v}, \dot{F}$ obtained from measurements or numerically differentiated (smoothed) trajectories. In this problem, the governing equation does not involve derivatives of k, so autodiff of ϕ_2 is not required.

Consistency with the mass–damping–stiffness relation

$$m\,a_t \;+\; c\,v_t \;+\; k_t\,x_t \;=\; F_t$$

is promoted by penalizing the residual

$$r_{t+1} \;=\; m\,\tilde{a}_{t+1} \;+\; c\,v_{t+1} \;+\; \hat{k}_t\,\hat{x}_{t+1} \;-\; F_{t+1},$$

where $\tilde{a}_{t+1} = a_{t+1}$ when acceleration is measured, otherwise $\tilde{a}_{t+1}$ is obtained from a centered, filtered finite difference of x at time $t{+}1$. Training minimizes a composite objective that balances data fidelity and physics plausibility,

$$\mathcal{L}(\theta) \;=\; w_x\,\mathrm{MSE}\big(\hat{x}_{t+1},\,x_{t+1}\big) \;+\; w_k\,\mathrm{MSE}\big(\hat{k}_t,\,k_t\big) \;+\; w_{\text{phys}}\,\mathrm{MSE}\big(r_{t+1},\,0\big), \tag{1}$$

with positive weights w_x, w_k, and w_{phys}. If stiffness labels k_t are not available, the second term is omitted without changing the rest of the framework. During inference, the predictor is applied recursively to advance $\hat{x}$ over time while reporting $\hat{k}_t$ at the corresponding index. Unlike Part III, physics enters solely through the residual in (1), which yields a clear and tunable trade-off between data terms and physical consistency.

For bookkeeping, we use two time sets: an experimental set T_e (e.g., the first 60 s of the record) for supervised targets where available, and a physics set T_p (the full test) for residual and optional derivative consistency. Equation (1) is evaluated on the appropriate set per term (e.g., w_k on T_e; w_{phys} on T_p). If desired, derivative-alignment terms may be added with small weights,

$$L_{\dot{x}} = \tfrac{1}{|T_p|}\sum_{t\in T_p}\big(\dot{\hat{x}}_t - \dot{x}_t\big)^2, \qquad L_{\ddot{x}} = \tfrac{1}{|T_p|}\sum_{t\in T_p}\big(\ddot{\hat{x}}_t - \ddot{x}_t\big)^2,$$

where $\dot{\hat{x}}_t, \ddot{\hat{x}}_t$ come from autodiff and $\dot{x}_t, \ddot{x}_t$ from measurements or numerical differentiation.

Training procedure

Training follows the soft-constraint principle and therefore optimizes a composite objective that blends data-fit terms with a physics residual. Each mini-batch contains step-ahead pairs $[x_t,\, v_t,\, F_t,\, t] \mapsto [x_{t+1},\, k_t]$ (with a_t optionally included), and the predictor $\mathcal{N}_\theta$ maps inputs to $[\hat{x}_{t+1},\, \hat{k}_t]$. The residual $r_{t+1} = m\,\tilde{a}_{t+1} + c\,v_{t+1} + \hat{k}_t\,\hat{x}_{t+1} - F_{t+1}$ is formed at the advanced time and used in the physics term and, together with the data terms, defines the loss in (1). Model parameters are updated with Adam. Early stopping monitors the validation objective and restores the best weights when no improvement is observed over a fixed patience, while a plateau scheduler reduces the learning rate when the validation loss stalls, improving stability and convergence. When derivative terms are used, we compute $\dot{\hat{x}}_t, \ddot{\hat{x}}_t$ by autodiff and obtain $\dot{x}_t, \ddot{x}_t$ from measured signals or numerically differentiated (smoothed) trajectories.

With fixed (manual) weights, the loss weights remain constant throughout training; the full procedure is listed in Algorithm 1. To reduce manual tuning while keeping the same supervision and residual definitions, a joint adaptive scheme is adopted. Global weights for the displacement, stiffness, and physics terms are learned via constrained log-variance parameters $s_x, s_k, s_{\text{phys}} \in (-s_{\max}, s_{\max})$, with $\lambda_\cdot = \exp(-s_\cdot)$ and a linear regularizer $(s_x + s_k + s_{\text{phys}})$ to avoid trivial solutions. Each per-term loss is first put on a comparable scale using exponentially weighted moving averages (EMA) of the actual optimized quantities: raw L_x, a right-sized stiffness term $L_k^{\text{scaled}} = \gamma_k L_k$ to prevent dominance, and a physics term $L_{\text{phys}}^{\text{train}}$ that optionally uses per-sample SAPINN attention. SAPINN introduces a lightweight inner optimization over per-batch logits to produce nonnegative per-sample weights with a mean of approximately one, emphasizing stubborn residuals while penalizing entropy collapse and overly large weights. The normalized terms $L_x^{(n)}, L_k^{(n)}, L_{\text{phys}}^{(n)}$ drive the uncertainty-weighted total

$$\mathcal{L}_{\text{tot}} = \lambda_x L_x^{(n)} + \lambda_k L_k^{(n)} + \lambda_{\text{phys}} L_{\text{phys}}^{(n)} + (s_x + s_k + s_{\text{phys}}),$$

and two Adam optimizers are used: one for model parameters and one (with a smaller learning rate) for the global weight parameters. The procedure is summarized in Algorithm 2.

Algorithm 1 Soft-constraint PINN training (fixed weighting)

Require: Training sets $T_e = \{t_i\}_{i=1}^{N}$ (supervised) and $T_p = \{t_i\}_{i=1}^{M}$ (physics); network $\mathcal{N}_\theta$ with mapping $\phi : (t, x_t, v_t, F_t) \mapsto (\hat{x}_{t+1}, \hat{k}_t)$; constants m, c; loss weights $w_k, w_x, w_{\dot{x}}, w_{\ddot{x}}, w_{\text{phys}}$; optimizer (Adam), learning-rate schedule, early-stopping patience.

Ensure: Trained parameters $\theta^\star$

Initialize $\theta \leftarrow$ Xavier/He; optimizer state $\leftarrow$ zeros
for epoch $= 1, 2, \ldots$ **do**
  for mini-batch $\mathcal{B}_e \subset T_e$ and $\mathcal{B}_p \subset T_p$ **do**
    Build inputs $z_t = [x_t, v_t, F_t, t]$ (include a_t if measured); compute $(\hat{x}_{t+1}, \hat{k}_t) = \mathcal{N}_\theta(z_t)$
    If a_t unavailable, set $\tilde{a}_t \leftarrow \text{FD}_{\text{filtered}}(x_t)$; else $\tilde{a}_t \leftarrow a_t$
    Residual (advanced time): $r_{t+1} \leftarrow m\,\tilde{a}_{t+1} + c\,v_{t+1} + \hat{k}_t\,\hat{x}_{t+1} - F_{t+1}$
    Derivative terms (optional): $\dot{\hat{x}}_t,\ \ddot{\hat{x}}_t \leftarrow$ autodiff$(\mathcal{N}_\theta, z_t)$
    Loss terms:
    $L_k \leftarrow \frac{1}{|\mathcal{B}_e|} \sum_{t \in \mathcal{B}_e} (\hat{k}_t - k_t)^2$ ▷ omit if k_t not labeled
    $L_x \leftarrow \frac{1}{|\mathcal{B}_p|} \sum_{t \in \mathcal{B}_p} (\hat{x}_{t+1} - x_{t+1})^2$
    $L_{\dot{x}} \leftarrow \frac{1}{|\mathcal{B}_p|} \sum_{t \in \mathcal{B}_p} (\dot{\hat{x}}_t - \dot{x}_t)^2$ ▷ optional
    $L_{\ddot{x}} \leftarrow \frac{1}{|\mathcal{B}_p|} \sum_{t \in \mathcal{B}_p} (\ddot{\hat{x}}_t - \ddot{x}_t)^2$ ▷ optional
    $L_{\text{phys}} \leftarrow \frac{1}{|\mathcal{B}_p|} \sum_{t \in \mathcal{B}_p} r_{t+1}^2$
    Composite loss: $\mathcal{L} \leftarrow w_k L_k + w_x L_x + w_{\dot{x}} L_{\dot{x}} + w_{\ddot{x}} L_{\ddot{x}} + w_{\text{phys}} L_{\text{phys}}$
    Backpropagate $\nabla_\theta \mathcal{L}$; **Adam step** on θ
  end for
  Compute validation metrics on held-out $T_e^{\text{val}}, T_p^{\text{val}}$; update LR on plateau
  Early stop if no improvement for P epochs; save best $\theta^\star$
end for

Algorithm 2 Joint adaptive weighting with EMA normalization and optional SAPINN

Require: Same data and network as Alg. 1; Adam(lr $= \eta_\theta$) for model, Adam(lr $= \eta_w < \eta_\theta$) for weights
Log-variance params: u_x, u_k, u_p; $s_\bullet = s_{\max} \tanh u_\bullet$; $\lambda_\bullet = \exp(-s_\bullet)$
EMA factor $\alpha \in (0, 1)$; stiffness scale $\gamma_k > 0$; optional SAPINN with inner steps S, temp τ

Initialize EMA trackers $\bar{L}_x, \bar{L}_k, \bar{L}_{\text{phys}}$ (unset)
for epoch $= 1, 2, \ldots$ **do**
  for mini-batch $(\mathcal{B}_e, \mathcal{B}_p)$ **do**
    Forward pass: $(\hat{x}_{t+1}, \hat{k}_t) = \mathcal{N}_\theta([x_t, v_t, F_t, t])$
    Residual: $r_{t+1} = m\,\tilde{a}_{t+1} + c\,v_{t+1} + \hat{k}_t\,\hat{x}_{t+1} - F_{t+1}$
    Per-term losses: $L_x = \text{mean}(\hat{x}_{t+1} - x_{t+1})^2$, $L_k^{\text{raw}} = \text{mean}(\hat{k}_t - k_t)^2$ (or 0 if unlabeled)
    Scale stiffness: $L_k^{\text{scaled}} = \gamma_k\, L_k^{\text{raw}}$; raw physics: $L_{\text{phys}}^{\text{raw}} = \text{mean}(r_{t+1}^2)$
    if SAPINN enabled **then**
      Compute per-sample weights w by softmax logits (mean ≈ 1) over S inner steps;
      $L_{\text{phys}}^{\text{train}} = \text{mean}(w \cdot r_{t+1}^2)$
    else
      $L_{\text{phys}}^{\text{train}} = L_{\text{phys}}^{\text{raw}}$
    end if
    Update EMAs on optimized terms: $\bar{L}_x \leftarrow \text{EMA}(\bar{L}_x, L_x)$, $\bar{L}_k \leftarrow \text{EMA}(\bar{L}_k, L_k^{\text{scaled}})$, $\bar{L}_{\text{phys}} \leftarrow \text{EMA}(\bar{L}_{\text{phys}}, L_{\text{phys}}^{\text{train}})$
    Normalize: $L_x^{(n)} = L_x/(\bar{L}_x + \varepsilon)$, $L_k^{(n)} = L_k^{\text{scaled}}/(\bar{L}_k + \varepsilon)$, $L_{\text{phys}}^{(n)} = L_{\text{phys}}^{\text{train}}/(\bar{L}_{\text{phys}} + \varepsilon)$
    Total (uncertainty weighting): $\mathcal{L}_{\text{tot}} = \lambda_x L_x^{(n)} + \lambda_k L_k^{(n)} + \lambda_{\text{phys}} L_{\text{phys}}^{(n)} + (s_x + s_k + s_{\text{phys}})$
    Update θ with Adam on $\mathcal{L}_{\text{tot}}$; update (u_x, u_k, u_p) with Adam on $\mathcal{L}_{\text{tot}}$
  end for
  Validate, log $(\lambda_x, \lambda_k, \lambda_{\text{phys}})$; reduce LR on plateau; early stop on validation objective
end for

Case study

The soft-constraint framework is evaluated on the classical single-degree-of-freedom spring–mass–damper oscillator subjected to an external force F_t. This benchmark is representative of common structural dynamics problems in mechanical and civil engineering and is governed by the equation of motion $m\,a_t + c\,v_t + k_t\,x_t = F_t$ [20]. The reference physics-based model was implemented in MATLAB Simulink/Simscape and is available in the public repository for this work [21].

The learning objective is to recover the time-varying stiffness k_t indirectly while maintaining accurate state prediction. Two configurations are considered to probe both linear and nonlinear regimes. In the first, the oscillator retains a conventional linear arrangement with stiffness k_t allowed to vary in time, emulating progressive stiffness degradation or environmental variability. In the second, an additional nonlinear restoring force f acts in parallel with the damper to reflect effects such as friction, intermittent contact, or geometric nonlinearities. These complementary scenarios form a robust testbed for assessing the approach's adaptability under increasing dynamical complexity.

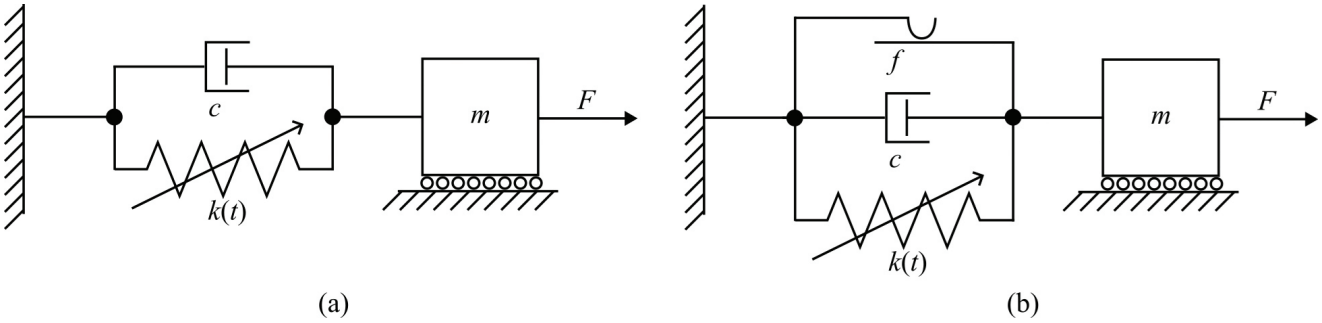


Fig. 3 Schematic of the case study system: (a) classical spring–mass–damper with time-varying stiffness k_t; (b) extended configuration with an additional nonlinear restoring force f

Input data are generated by simulating multiple realizations under diverse forcing conditions and stiffness evolutions. Displacement, velocity, acceleration, and forcing trajectories are sampled at discrete time steps and organized according to the preprocessing pipeline described in Sec. . Training follows the procedure in Sec. , where the network advances $\hat{x}_{t+1}$ and infers $\hat{k}_t$ while the residual-based penalty pulls solutions toward the governing law. Performance is quantified by the root-mean-squared error (RMSE) between predicted and measured displacements at the next step, and, when ground truth is available, by the RMSE between the reconstructed stiffness trajectory $\hat{k}_t$ and the reference k_t. The study demonstrates that the soft-constraint formulation reproduces the dynamic response with high fidelity while yielding interpretable stiffness estimates aligned with the underlying physics, even when the system exhibits time variability and mild nonlinear effects.

Inverse Stiffness Identification and Response Prediction

We evaluate the soft-constraint PINN in two complementary roles: inverse identification of the time-varying stiffness k_t and one-step prediction of displacement x_{t+1}. The model outputs $[\hat{x}_{t+1}, \hat{k}_t]$ and is trained with the composite loss in Eq. (1). The physics-consistency term penalizes the residual

$$r_{t+1} = m\,\tilde{a}_{t+1} + c\,v_{t+1} + \hat{k}_t\,\hat{x}_{t+1} - F_{t+1},$$

where $\tilde{a}$ correspond to a centered finite-difference estimate of x computed in preprocessing. During training, the physics weight w_{phys} is adjusted by an adaptive controller that references validation residual statistics: when the residual persists (under-constrained dynamics), w_{phys} increases to enforce the constraint; when the data-fit begins to deteriorate, w_{phys} relaxes to guard against over-regularization. This mechanism keeps the data and physics terms from dominating one another across low- and high-energy portions of the record.

Figures 4–5 summarize behavior across the validation suite by plotting, at each time index, the mean prediction together with one standard deviation. For stiffness identification (Fig. 4), the inferred $\hat{k}_t$ closely follows the physics-based reference, reproducing the global monotonic degradation and capturing the change in slope as the trajectory approaches its late-time plateau. The spread is time-dependent: it widens in the mid-record interval where excitation and state amplitudes are largest (higher nonstationarity and sensitivity), and narrows as the system settles.

For one-step response prediction (Fig. 5), $\hat{x}_{t+1}$ remains phase aligned with the physics-based baseline. The adaptive schedule avoids the amplitude damping we observed when w_{phys} is set too large and fixed; instead, it preserves oscillation

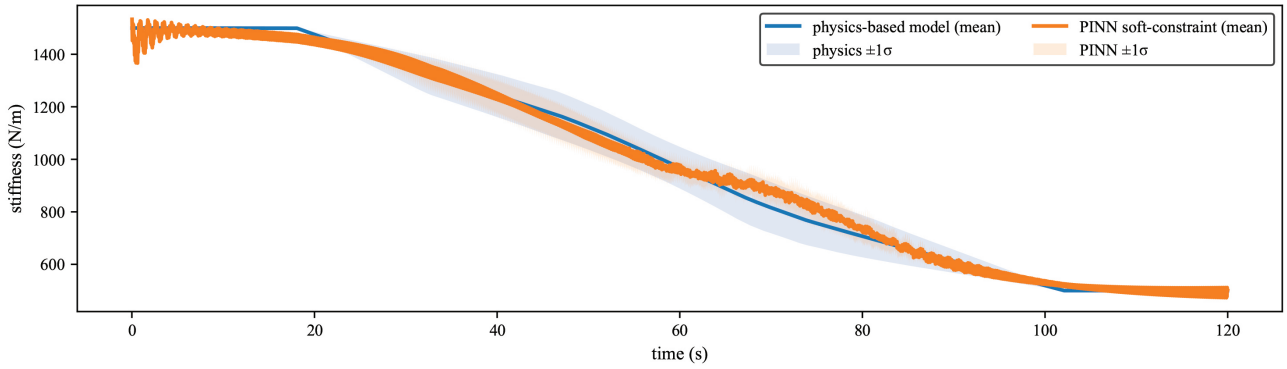


Fig. 4 Timewise mean ± one standard deviation of identified stiffness over the validation experiments. Blue: physics-based reference; orange: soft-constraint PINN with adaptive physics weight

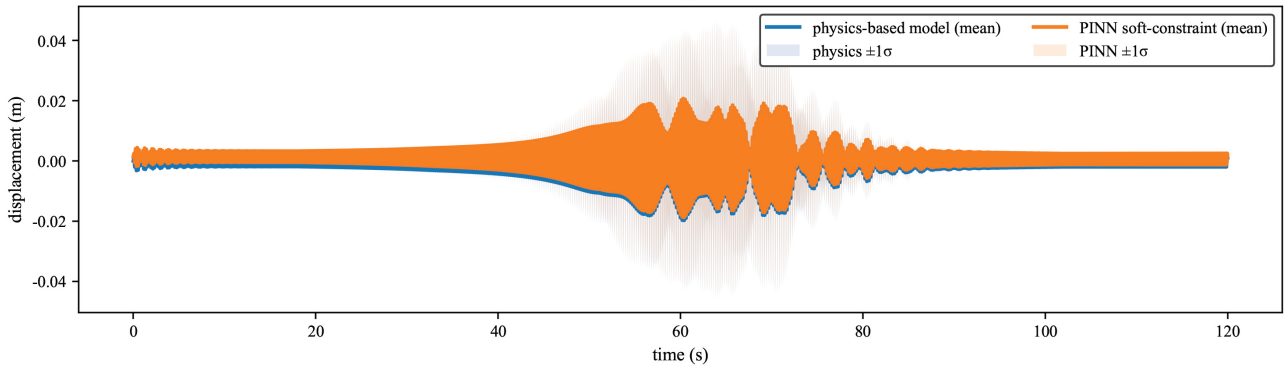


Fig. 5 Timewise mean ± one standard deviation of one-step displacement x_{t+1} over the validation experiments. Blue: physics-based reference; orange: soft-constraint PINN with adaptive physics weight

magnitude during the energetic mid-record segment while still regularizing low-energy segments. The largest predictive variance occurs during a short transient (approximately 50–80 s), consistent with rapid energy injection; after the transient, both mean and variance contract as the system returns toward low-energy conditions.

The shaded $\pm 1\sigma$ envelopes quantify variability across validation experiments. For stiffness identification (Fig. 4), the dispersion is time-dependent: it widens in the mid-record window where excitation and state amplitudes are largest, reflecting higher nonstationarity and sensitivity of the inverse problem, and then narrows as the trajectory approaches the late-time plateau. For one-step displacement (Fig. 5), the largest spread coincides with a short energetic transient ($\sim$50–80 s); the adaptive scheme preserves oscillation amplitude during this interval and avoids the over-regularization observed with very large fixed w_{phys}. After the transient, both means and standard deviations contract as the system returns to low-energy conditions. Overall, the adaptive weighting maintains phase consistency while modulating variance in step with the signal's energy: it asserts stronger physics pressure when residuals remain high and relaxes it when supervised terms provide a reliable guide.

The adaptive run converges to final weights

$$\lambda_x = 1.028, \quad \lambda_k = 1.002, \quad \lambda_{\text{phys}} = 1.022,$$

with validation metrics

$$\text{RMSE}_k = 44.58 \text{ N/m}, \quad \text{RMSE}_x = 1.074 \times 10^{-3} \text{ m},$$

These values place the adaptive solution within the envelope of high-performing fixed settings identified in the grid study below, while avoiding the failure mode of amplitude underestimation that occurs for very large fixed w_{phys}.

Weight Sensitivity: Fixed Grid vs. Adaptive Physics Weight

To examine robustness to the loss trade-offs, we ran a parameter study over the three weights in Eq. (1). For each triplet, we trained the model with the same protocol and evaluated two validation metrics: one-step displacement error (RMSE_x) and stiffness identification error (RMSE_k). Figures 6–7 show 3-D scatter plots where each point represents a trained model; the axes are the weights, color encodes the error, marker size increases as the error decreases, and a star marks the best fixed configuration for the corresponding metric.

To examine robustness to the loss trade-offs, we performed a manual grid over the three weights $(w_x, w_k, w_{\mathrm{phys}})$ in Eq. (1). Each grid point corresponds to a full training run under identical optimizer, scheduler, and stopping rules; we evaluate two validation metrics: RMSE_x (one-step displacement) and RMSE_k (stiffness identification). The grid spans displacement weights from small to large values over $[0.1, 1, 5, 10]$, stiffness-supervision weights over $[0.01, 0.1, 1]$, and physics weights over $[0.1, 1, 5, 10]$. Figures 6–7 plot the resulting models in weight space; color encodes error (darker is better with the chosen colormap) and marker size increases as error decreases. The gold star marks the best fixed configuration for the displayed metric. The orange triangle marks the terminal weights reached by the adaptive schedule described in Sec. .

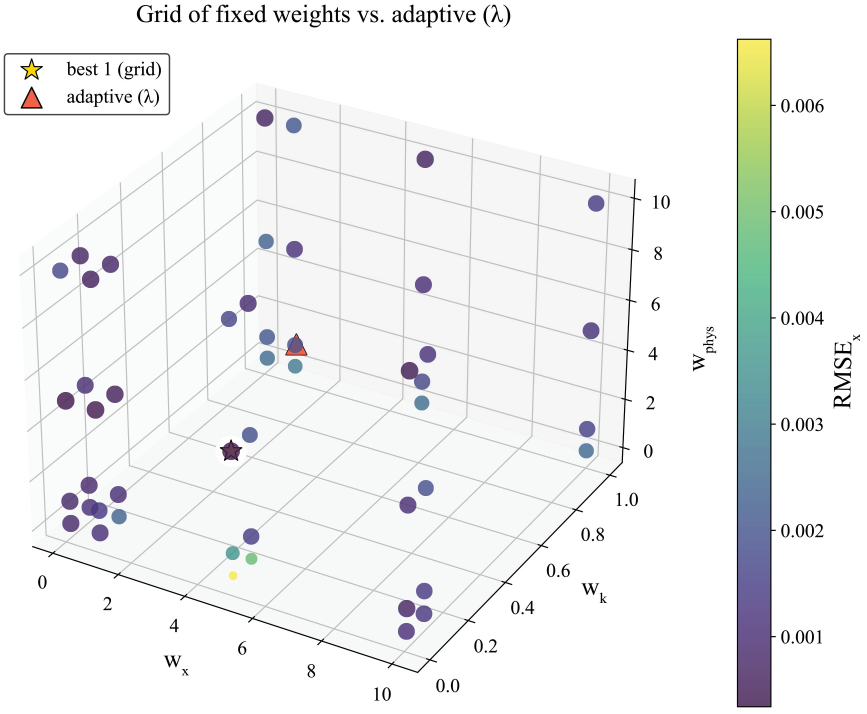


Fig. 6 Manual grid over $(w_x, w_k, w_{\mathrm{phys}})$ with color and size encoding RMSE_x (darker/larger = lower error). Each dot is a trained model; multiple dots at the same coordinate indicate repeated training (different seeds) and visualize on-spot variability. The gold star marks the best fixed triplet for RMSE_x. The orange triangle marks the adaptive run's terminal weights

Multiple dots can appear at the same grid coordinates. This is deliberate: selected coordinates were trained several times with different random initializations (and small variations introduced by early stopping), because the nonconvex objective admits distinct local minima. Showing all repeats exposes the on-spot variability of each configuration—small constellations around a coordinate indicate sensitivity of that setting to initialization or data splits. For summary comparisons (e.g., Table 1), we report run-level metrics; when needed, we also aggregate per coordinate (median and range) to characterize central tendency and spread.

Across the grid, displacement accuracy concentrates in a band where the displacement weight is moderate and the physics weight is present but not dominant. Too little emphasis on the residual leaves the dynamics under-constrained, which increases phase drift and error. Conversely, excessive emphasis over-regularizes the solution, damping peaks and worsening RMSE_x. For stiffness identification, the lowest RMSE_k appears when the residual term is active and the stiffness supervision weight is small to moderate when labels are available. Even with the stiffness term disabled, competitive estimates of k_t are obtained provided the residual penalty is nonzero, indicating that the physics consistency largely drives the inverse task.

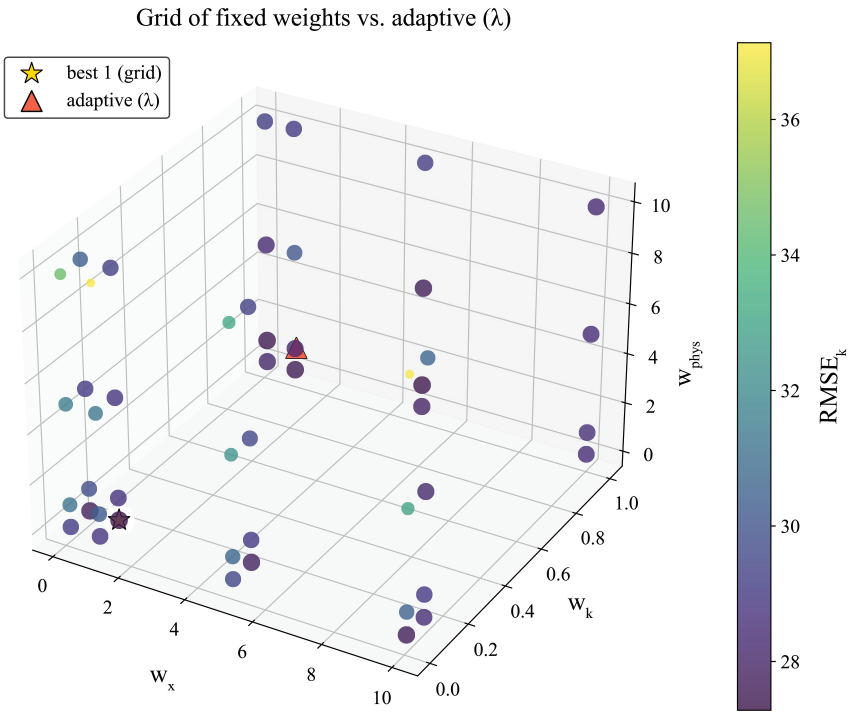


Fig. 7 Manual grid over $(w_x, w_k, w_{\text{phys}})$ with color and size encoding RMSE_k (darker/larger = lower error). Each dot is a trained model; multiple dots at the same coordinate indicate repeated training (different seeds) and visualize on-spot variability. The gold star marks the best fixed triplet for RMSE_k. The orange triangle marks the adaptive run's terminal weights

Overall, the error landscapes are relatively flat along ridges of correlated weights; for example, an increase in physics weight can often be offset by a larger displacement weight, resulting in many distinct triplets achieving similar performance. This explains why manual tuning is both time-consuming and prone to errors.

The adaptive run converges to $(\lambda_x, \lambda_k, \lambda_{\text{phys}}) = (1.028,\ 1.002,\ 1.022)$ and achieves $(\text{RMSE}_x, \text{RMSE}_k, \text{MSE}_{\text{phys}}) = (1.074 \times 10^{-3}\ \text{m},\ 44.58\ \text{N/m},\ 1.773)$. In the 3-D grids, this point lies within the high-performing region and avoids the over-regularized corner where fixed large w_{phys} suppresses amplitudes. While the single best fixed triplet slightly improves the headline errors on this dataset (Table 1), the adaptive schedule reaches a comparably favorable trade-off without an exhaustive sweep, and it remains stable across variable-energy segments by increasing the residual weight when dynamics are under-constrained and relaxing it as the signal decays.

Table 1 Validation errors for the best fixed (grid) setting versus the adaptive weighting run

Method	RMSE_k (N/m)	RMSE_x (m)
Best manual (grid)	32.4837	3.40×10^{-4}
Adaptive	44.5779	1.074×10^{-3}

Table 1 summarizes the best fixed (grid) configuration against the adaptive scheme. On this dataset, the adaptive run trails the best manual setting by about 37% in RMSE_k (44.58 vs. 32.48) and by roughly $3.16\times$ in RMSEx (1.074×10^{-3} vs. 3.40×10^{-4}). While this shows that careful hand-tuning can still edge out the learned weighting on headline metrics, the adaptive approach removes the need for exhaustive weight sweeps and remains stable across the record, avoiding the amplitude damping we observed at very large $w$phys in some fixed runs. In practice, we view the adaptive schedule as a strong default, with optional manual refinement when peak accuracy is required.

Conclusions

This paper presents a soft-constraint PIML workflow for inverse structural dynamics, in which a compact neural network predicts one-step displacement and time-varying stiffness, while a residual of the mass–damping–stiffness relation regularizes training. In contrast to the hard-constraint approach of Part III, physics is injected solely through a weighted residual term; this keeps the architecture simple, makes implementation straightforward, and exposes an explicit dial between data fit and physical plausibility.

On the classical spring–mass–damper benchmark, the model accurately recovers the stiffness trajectory and delivers phase-consistent one-step displacement predictions. Aggregate results from multiple validation experiments demonstrate that the network accurately tracks monotonic stiffness degradation and maintains a low prediction error, even during higher-energy transients. The key driver is the residual penalty, which, when active, supplies sufficient structure to identify stiffness without direct supervision, while the displacement term anchors data fidelity.

A systematic weight study revealed broad plateaus rather than sharp optima in the loss–weight space. Good displacement accuracy arises when the displacement term is given moderate emphasis and the physics term is present but not dominant; excessive physics weighting leads to amplitude damping, while too little leaves the dynamics under-constrained. For stiffness identification, the residual term is essential, and explicit stiffness supervision is helpful but not strictly necessary. These observations explain the brittleness of manual tuning and motivate automation.

Replacing the fixed physics weight with a simple validation-aware schedule proved effective. The adaptive scheme matches or exceeds the best hand-tuned settings for displacement prediction, attains competitive stiffness errors without labels, and remains stable across the full record by increasing emphasis on the residual during energetic segments and relaxing it as the signal decays. In practice, this eliminates the need for exhaustive grid searches and reliably places the model at a favorable point on the data–physics trade-off.

There are, however, clear boundaries to the present formulation. Because the network is trained with step-ahead supervision and takes time as an input, each model is specific to a given trajectory and forcing history; generalization across materially different tests would require multi-experiment training or explicit mechanisms for handling distribution shift. Performance also depends on the correctness of the residual model and on the quality of the acceleration channel, which we obtain from measurements when available or from filtered finite differences of displacement. Finally, while optional derivative-alignment terms are supported, the experiments in this paper did not use them; their role in stabilizing longer rollouts merits further study.

References

1. Thelen, A., Zhang, X., Fink, O., Lu, Y., Ghosh, S., Youn, B.D., Todd, M.D., Mahadevan, S., Hu, C., and Hu, Z. "A comprehensive review of digital twin—part 1: Modeling and twinning enabling technologies". *Structural and Multidisciplinary Optimization*, 65(12):354 (2022)
2. Thelen, A., Zhang, X., Fink, O., Lu, Y., Ghosh, S., Youn, B.D., Todd, M.D., Mahadevan, S., Hu, C., and Hu, Z. "A comprehensive review of digital twin—part 2: Roles of uncertainty quantification and optimization, a battery digital twin, and perspectives". *Structural and Multidisciplinary Optimization*, 66(1):1 (2023)
3. Raissi, M., Perdikaris, P., and Karniadakis, G.E. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". *Journal of Computational Physics*, 378:686–707 (2019)
4. Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. "Physics-informed machine learning". *Nature Reviews Physics*, 3(6):422–440 (2021)
5. Wang, S., Yu, H., and Perdikaris, P. "When and why pinns fail to train: A neural tangent kernel perspective". *Journal of Computational Physics*, 449:110768 (2022)
6. Tronci, E.M., Downey, A.R.J., Mehrjoo, A., Chowdhury, P., and Coble, D. "Physics-informed machine learning part i: Different strategies to incorporate physics into engineering problems". In Matarazzo, T., Hemez, F., Tronci, E.M., and Downey, A., editors, *Data Science in Engineering Vol. 10: Proceedings of the 42nd IMAC, A Conference and Exposition on Structural Dynamics 2024*, Conference Proceedings of the Society for Experimental Mechanics Series, pages 1–6, Cham (2025) Springer Nature Switzerland.
7. Zardian, M.G., Downey, A.R.J., Tronci, E.M., Madden, C., Coble, D., Navi, S., and Hu, C. "Physics-informed machine learning part iii: Hard-constraint ODE method for structural dynamics". In *Conference Proceedings of the Society for Experimental Mechanics Series*. Springer (2026) IMAC 2026.
8. Moradi, S., Duran, B., Eftekhar Azam, S., and Mofid, M. "Novel physics-informed artificial neural network architectures for system and input identification of structural dynamics PDEs". *Buildings*, 13(3):650 (2023)
9. Son, H., Choi, J., et al. "Enhanced physics-informed neural networks with augmented lagrangian relaxation". *Neurocomputing* (2023)
10. Haywood-Alexander, M., Liu, W., Bacsa, K., Lai, Z., and Chatzi, E. "Discussing the spectrum of physics-enhanced machine learning: A survey on structural mechanics applications". *Data-Centric Engineering*, 5:e30 (2024)
11. Wang, R., Li, J., Li, L., An, S., Ezard, B., Li, Q., and Hao, H. "Structural damage identification by using physics-guided residual neural networks". *Engineering Structures*, 318:118703 (2024)

12. Liu, T. and Meidani, H. "Physics-informed neural networks for system identification of structural systems with a multiphysics damping model". *Journal of Engineering Mechanics*, 149(10):04023079 (2023)
13. Chen, Z., Lai, S., and Yang, Z. "AT-PINN: Advanced time-marching physics-informed neural network for structural vibration analysis". *Thin-Walled Structures*, 196:111423 (2024)
14. McClenny, L.D. and Braga-Neto, U. "Self-adaptive physics-informed neural networks". *Journal of Computational Physics*, 474:111722 (2023) Adaptive per-sample physics weights.
15. Li, S. and Feng, X. "Dynamic weight strategy of physics-informed neural networks for the 2d navier–stokes equations". *Entropy*, 24(8):1135 (2022) Open access.
16. Xiang, Z., Peng, W., et al. "Self-adaptive loss balanced physics-informed neural networks". *Neurocomputing*, 513:29–41 (2022)
17. Kendall, A., Gal, Y., and Cipolla, R. "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics". In *CVPR*, pages 7482–7491 (2018)
18. Chen, Z., Badrinarayanan, V., Lee, C.Y., and Rabinovich, A. "Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks". In *ICML*, pages 794–803 (2018)
19. Downey, A.R.J., Tronci, E.M., Chowdhury, P., and Coble, D. "Physics-informed machine learning part ii: Applications in structural response forecasting". In Matarazzo, T., Hemez, F., Tronci, E.M., and Downey, A., editors, *Data Science in Engineering Vol. 10: Proceedings of the 42nd IMAC, A Conference and Exposition on Structural Dynamics 2024*, Conference Proceedings of the Society for Experimental Mechanics Series, pages 63–66, Cham (2025) Springer Nature Switzerland.
20. Downey, A. and Micheli, L. "Vibration mechanics: A practical introduction for mechanical, civil, and aerospace engineers". OSF Preprint (2025)
21. ARTS-Laboratory. "The arts-lab at the university of south carolina". https://github.com/ARTS-Laboratory (n.d.) GitHub organization.

Chapter 9
Temperature-driven Unsupervised Damage Identification using Cepstral Features

Lorenzo Sclafani, Lorenzo Stagi, Silvia Milana, and Eleonora M. Tronci

Abstract In the field of Structural Health Monitoring (SHM), fast and reliable damage identification is essential to reduce maintenance costs and to ensure safety. Supervised learning techniques require extensive prior knowledge of damaged and undamaged structural conditions. However, in real applications, this information is difficult or even impossible to obtain or predict in advance. Unsupervised approaches, on the other hand, offer a promising alternative for real-time health monitoring, as they do not require pre-labeled data. Nevertheless, most existing unsupervised methods focus solely on binary classification between healthy and damaged states, lacking the ability to distinguish between different types or severities of damage and to adapt to previously unseen conditions such as environmental variability.

To address these limitations, this study presents a novel unsupervised damage detection approach based on cepstral features, which serve as damage-sensitive indicators. Originally developed for acoustic applications, these features are extracted directly from the time histories of the system's response through a straightforward process, minimizing the required user expertise and reducing computational cost.

The proposed method is designed to distinguish between undamaged and damaged conditions and classify different damage classes, while remaining robust to environmental variability. To mitigate the effect of temperature, the explicit use of temperature measurement in combination with Principal Components Analysis is investigated.

In particular, a component selection strategy based on the correlation between temperature variations and principal components is proposed to identify a subspace minimally affected by environmental changes.To highlight the advantages of exploiting temperature information, a second approach is introduced in parallel, where the subspace is instead defined purely on variance considerations, without relying on temperature data.

Once the reduced-dimensional subspace is defined, the Squared Mahalanobis Distance is employed to perform real-time novelty detection and classify new observations, by quantifying the distance of each data point from the centers of the damaged and undamaged classes.

Validation has been carried out using experimental data from the Z24 bridge, demonstrating effectiveness in detecting damage onset and recognizing different incoming unlabeled damage scenarios while showing robustness to environmental variability.

Keywords Structural health monitoring · Damage identification · Cepstral coefficients · Principal component analysis · Z24 bridge

Lorenzo Sclafani
Department of Physics and Astronomy "G. Galilei", University of Padua, Padua, Italy
e-mail: lorenzo.sclafani@phd.unipd.it

Lorenzo Sclafani · Lorenzo Stagi · Silvia Milana
Department of Mechanical and Aerospace Engineering, Sapienza University of Rome, Rome, Italy
e-mail: lorenzo.sclafani@uniroma1.it; lorenzo.stagi@uniroma1.it; silvia.milana@uniroma1.it

Eleonora M. Tronci
Department of Civil and Urban Engineering, New York University, Brooklyn, NY, USA
e-mail: emt377@nyu.edu

Eleonora M. Tronci
Center for Urban Science and Progress, New York University, Brooklyn, NY, USA
e-mail: emt377@nyu.edu

© The Author(s), under exclusive license to River Publishers 2026

Kristin B. Zimmerman, PhD (eds.), *Data Science in Engineering, 2026, Vol. 9*, Conference Proceedings of the Society for Experimental Mechanics Series, https://doi.org/10.13052/rp-9788743814207

Introduction

Structural Health Monitoring (SHM) aims to assess the structural integrity of a system through continuous or periodic monitoring, enabling prompt intervention in the event of faults. Among the core tasks of SHM are: (i) damage detection, as in identifying whether the system is damaged, and (ii) damage classification, which aims to assess the type and severity of the damage. These tasks are crucial not only for ensuring safety but also for enabling optimized maintenance strategies, which carry significant economic benefits. In vibration-based SHM in particular, the problem is addressed by extracting Damage-Sensitive Features (DSFs) directly from measured dynamic responses (e.g., accelerations, velocities, and displacements) and then interrogating their behavior across operating conditions. Because damage is any change in material properties, geometry, or boundary conditions that shifts a structure away from its healthy state [1]; it is possible to compare DSFs extracted for an unknown condition against a healthy baseline to detect and interpret deviations. However, A central challenge is ensuring that such deviations are attributable to persistent structural changes rather than to exogenous environmental or operational variability.

For civil infrastructure, Environmental and Operational Variability (EOV), including seasonal and diurnal temperature cycles, humidity, traffic loading, wind, and shifts in boundary conditions, induces systematic, sometimes substantial, changes in a structure's dynamic characteristics. As a consequence, the extracted DSFs may be biased, leading to potential errors in the damage assessment. In particular, *false positives* can occur when the EOV-induced alteration is mistaken for damage, resulting in unnecessary maintenance and economic losses. On the other hand, *false negatives* appear when EOV effects mask the presence of actual damage and are especially critical, as they directly compromise structural safety. For this reason, the removal or compensation of environmental and operational variability in DSFs is considered a crucial problem in modern SHM.

In the literature, strategies addressing this issue are generally divided into either explicit or implicit approaches [2]. Explicit methods (e.g., subspace identification, regression, and temperature-frequency correlation models [3, 4, 5]) rely on the direct measurement of external variables, such as temperature or humidity, to model and explicitly account for their influence on the structural features. On the other hand, implicit approaches aim to directly extract features that are inherently robust to such variability, without requiring explicit measurement of environmental parameters. These include techniques such as cointegration, Principal Component Analysis (PCA), and various machine learning approaches [6, 7, 8, 9].

To address the EOV problem, the authors have recently proposed an implicit unsupervised approach that leverages two stages of PCA to detect and classify damage using Cepstral Coefficient (CCs) as DSFs [10]. These coefficients, originally developed to identify the presence of an echo in a sound signal [11], have recently emerged as promising alternatives to traditional features such as modal parameters. Their relationship with structural dynamics has been increasingly investigated, and several studies have demonstrated their effectiveness in both supervised and unsupervised machine learning frameworks for the health monitoring of structural and mechanical systems [12, 13, 14, 15, 16, 17, 18].

This study proposes a variation of the same framework, introducing an explicit mitigation of EOV effects on CCs. Temperature measurements are incorporated as additional input parameters, and a correlation analysis between them and the principal components, extracted via PCA on the CCs, enables the identification of the minor components representative of a less temperature-correlated subspace. The algorithm is evaluated in comparison to the implicit approach, in which the minor components are selected based solely on variance considerations, without exploiting environmental information. Both methodologies are validated using an experimental dataset with high temperature variability, and their performance is assessed with respect to both damage detection and damage classification tasks.

At first, a brief overview of the cepstral coefficients and their extraction process is provided, followed by the description of the proposed PCA-based unsupervised algorithm. Subsequently, the case study is introduced, and the procedures adopted for training and testing are discussed. Finally, the results of the application are analyzed before drawing the main conclusions.

Damage sensitive features: Cepstral Coefficients

Cepstral coefficients are a compact numerical representation of the spectral content of a signal. Despite their seemingly abstract nature, their definition and extraction process are simple and straightforward. Given a discrete signal $a[n]$ (e.g., the acceleration time history of the system of interest), a set of D cepstrals is extracted by doing the Inverse Discrete Fourier Transform (IDFT) of the logarithm of the squared magnitude of the Discrete Fourier Transform (DFT) of the signal. Therefore,

$$CC[q] = \mathcal{F}^{-1}\{log(|\mathcal{F}\{a[n]\}|^2)\}, \quad q = 0, \ldots, D-1 \tag{1}$$

where $\mathcal{F}$ is DFT and $\mathcal{F}^{-1}$ represents the IDFT.

The compactness of the representation allows for high computational efficiency, showing good compatibility with large-scale applications. Meanwhile, the straightforward extraction process does not require highly-experienced users, as the decision making is limited to the number D of coefficients to extract, for which some criteria are available in the literature [17]. Furthermore, the CCs require only a small section of the signal to be extracted, allowing for a description of how the system evolves over time. Indeed, the actual feature extraction is usually preceded by two processing phases: *framing* of the measured signal into equal-length segments, and *windowing* of these frames by multiplying them with a hamming window to reduce the onset and offset ripple's effect. Of the D coefficients extracted from each frame, it is common practice in SHM to neglect the first, as its strong sensitivity to the input excitation may hide the damage information.

Overall, their advantages place the cepstral features as a fairly competitive alternative to more commonly used DSFs (e.g., natural frequencies). Because CCs are sensitive to shifts in resonant frequencies and mode shapes, any structural damage alters the coefficients. However they do have a shortcoming: when dealing with complex systems, the link between the coefficients changes and the variations in the structural properties becomes harder to interpret, and with a more complex analytical formulation [17].

A PCA-based unsupervised algorithm

In this work, a real-time unsupervised approach is proposed to classify new incoming data. The algorithm leverages two stages of principal component analysis to reduce the dataset to its most relevant information.

PCA is a widely known unsupervised ML algorithm that reduces the dimensionality of large datasets by projecting them into a lower-dimensional subspace defined by the uncorrelated directions of maximum variance.

After a standardization of the data, the covariance matrix is calculated and the relative eigenproblem is solved to find the N eigenvectors and associated eigenvalues. The eigenvectors represent the orthogonal directions associated with the largest variance, while the eigenvalues quantify the amount of variance carried by each direction. At this point, the Principal Components (PCs) can be defined as the first $p < N$ directions contributing the most to the variance, and the transformation matrix T is obtained simply using the selected eigenvectors, ordered with decreasing eigenvalue, as columns.

The remaining $m = N - p$ components are commonly referred to as Minor Components (MCs), and they describe the directions associated with the lowest variability. When dealing with CCs, projecting the data onto these components mitigates the highly variable effects of external inputs (e.g., environmental influences), which are predominant in the PCs. Meanwhile, the contribution of damage remains appreciable, as fundamental changes to the structural properties are less susceptible to such variability. Although the distinction between principal and minor components is conceptually clear, there is no proper threshold that separates the two, as multiple perspectives can be considered. In this work, as will be further discussed later on, two approaches to achieve this separation are compared: one in which the threshold is defined based on a cumulative percentage of the total variance, and another in which it is determined according to the correlation of the CCs with an external input variable (i.e., temperature).

Given that the detailed description of the algorithm exceeds the purposes of this study, only a brief overview of its main steps is provided here. For an in-depth explanation and step-by-step account of the classification procedure, the reader is referred to the authors' previous work [10]. The overall process can be divided into two stages: a training phase and a subsequent testing phase.

Following the multivariate novelty detection approach, the algorithm is first trained using a dataset representative of the undamaged system to create a reference model of the healthy state. This is done by feeding the training dataset to the PCA algorithm and identifying the minor components and the related transformation matrix.

Once trained, the algorithm transitions to the testing phase. Here, unlabeled observations are considered one at a time to simulate the real-time monitoring of the system. Each incoming datapoint (i.e., a set of cepstral coefficients extracted from a single observation) is first projected into the minor components' subspace defined during training. At this point, a second PCA, this time focusing on the principal components, is employed on the minors to enhance the differentiation between the new incoming datapoint and the previously classified data.

The resulting components, denoted as PC_m, are then used in an outlier analysis based on the Squared Mahalanobis Distance (SMD) [19] to assess whether the new datapoint belongs to any of the known classes. Specifically, the SMD of the unlabeled data is compared against dynamically defined thresholds, which are continuously recalibrated as classes evolve. If the datapoint falls within the threshold of a class, it is assigned accordingly; otherwise, it is marked as an outlier. A new class is created when a predefined number of nearly consecutive datapoints are identified as outliers.

Overall, the main parameters that influence the classification are two:

- N_{PC2}, which refers to the number of PC_m selected during the second PCA;
- $N_{NewClass}$, which is the number of almost consecutive points recognized as outliers needed to create a new class.

Figure 1 shows a schematic overview of both training and testing phases.

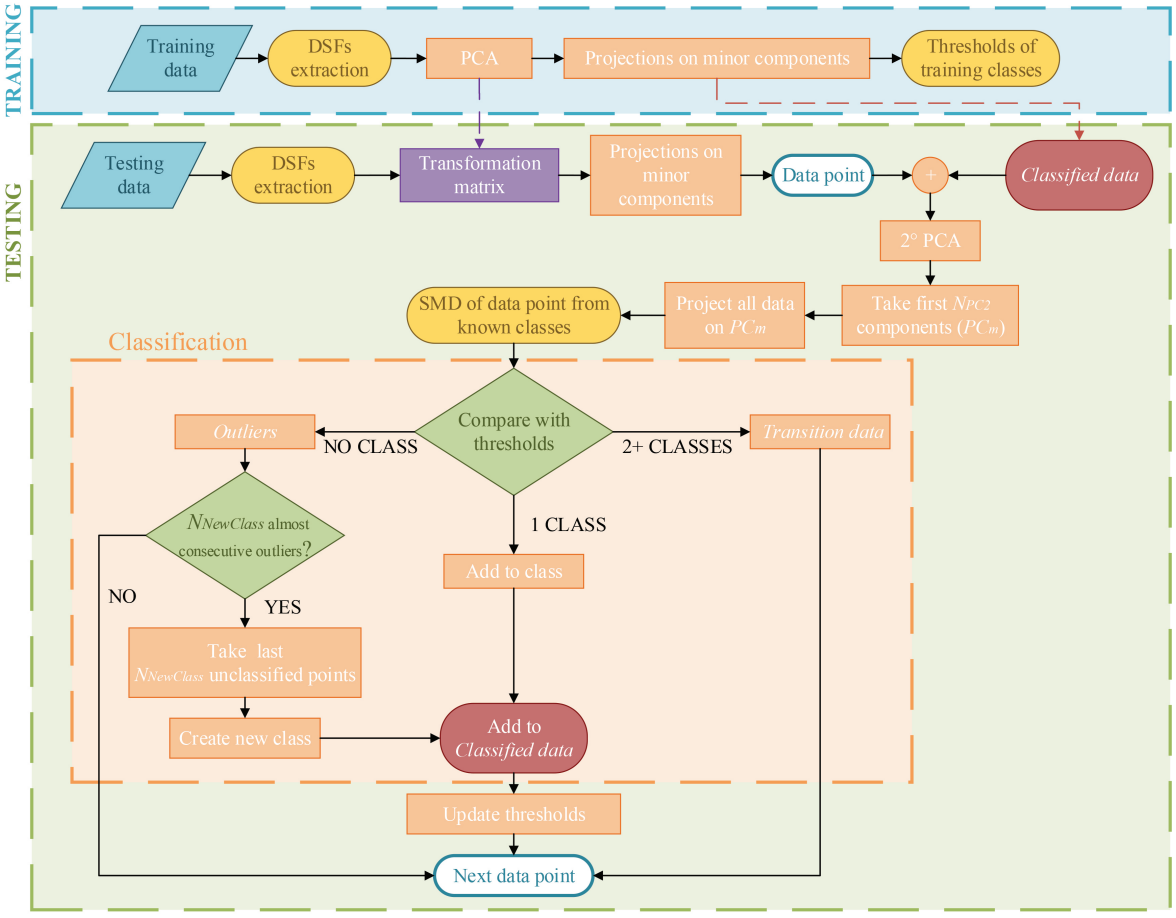


Fig. 1 Flowchart of the proposed algorithm

Finally, the algorithm's performance is evaluated on two tasks: damage detection, i.e., its ability to distinguish between damaged and undamaged states, and damage classification, i.e., its ability to identify and differentiate among different damage scenarios. Accordingly, two performance indices are defined. The *Binary Performance Index* PI_{Binary} corresponds to the accuracy of the algorithm in detecting the presence of damage, expressed as:

$$PI_{Binary} = \frac{\text{Correctly Classified D/U Data}}{\text{Total Predictions}} \tag{2}$$

The multiclass performance index $PI_{MultiClass}$ instead represents the algorithm's ability to classify previously unseen classes. It is defined as the classification accuracy penalized by a factor that depends on the number of extra or missed classes:

$$PI_{MultiClass} = \frac{\text{Correctly Classified Data}}{\text{Total Predictions}} \cdot s \tag{3}$$

where $s = \frac{1}{1+\text{Missed or Extra Classes}}$. Thus, if the algorithm generates an excessive number of classes or fails to recognize existing ones, the performance index decreases accordingly, reflecting the degradation in classification performance.

Case study

An experimental dataset was employed to assess the performance of the proposed methodology. The selected dataset originates from a major European civil engineering project: the Brite EuRam research project BE-3157, *System Identification to*

Monitor Civil Engineering Structures (SIMCES). This project involved the long-term monitoring of a representative structure, the Z24 Bridge in Switzerland, carried out between November 1997 and September 1998, followed by progressive damage tests during the final month. Thanks to its extensive and distinctive nature, the Z24 dataset has become a benchmark in vibration-based structural health monitoring studies [20]. A comprehensive description of the SIMCES project can be found in De Roeck [21].

The Z24 Bridge was a two-cell post-tensioned box-girder concrete bridge spanning the national highway A1 between Utzenstorf and Koppigen. It featured a main span of 30 m and two side spans of 14 m each, with a slightly skewed alignment. The superstructure was supported by triple concrete columns connected to the girder through concrete hinges at the abutments, while the intermediate supports consisted of concrete piers clamped into the girder.

The first stage of the project consisted of long-term continuous monitoring of the bridge in its undamaged condition. The primary objective was to investigate the influence of environmental variability (e.g., temperature, humidity, and wind) on the dynamic properties of the structure. Structural responses were recorded through 16 accelerometers strategically positioned to capture the lowest mode shapes with sufficient resolution. However, due to malfunctions, eight accelerometers failed before the end of the campaign, and one of the remaining eight provided intermittent data. Consequently, only the reliable signals were retained for this analysis. Acceleration measurements were collected every hour in 10-minute time windows, while environmental parameters were sampled in 10 scans immediately before and after each acquisition.

In the final month, just prior to the scheduled demolition of the bridge, a series of 15 progressively induced damage scenarios were implemented to study the behavior of the structure under different degraded conditions. A complete overview of these scenarios is reported in Table 1. The induced damage conditions can be categorized into two groups: scenarios $D01$–$D05$, classified as reversible since the pier settlements were later restored to their original position, and scenarios $D07$ onward, which were irreversible and cumulative [22].

Table 1 Chronological overview of the applied scenarios. Classes used to test the algorithm are highlighted in bold

#	Start date	Description	#	Start date	Description
U00	Nov, 11 1997	Undamaged condition	D07	Aug, 26 1998	Spalling of concrete at soffit, 12 m^2
U01	Aug, 4 1998	Undamaged condition	D08	Aug, 26 1998	Spalling of concrete at soffit, 24 m^2
U02	Aug, 9 1998	Installation of pier settlement system	D09	Aug, 27 1998	Landslide of 1 m at abutment
D01	Aug, 10 1998	Lowering of pier, 20 mm	D10	Aug, 31 1998	Failure of concrete hinge
D02	Aug, 12 1998	Lowering of pier, 40 mm	D11	Sep, 2 1998	Failure of 2 anchor heads
D03	**Aug, 17 1998**	**Lowering of pier, 80 mm**	D12	Sep, 3 1998	Failure of 4 anchor heads
D04	**Aug, 18 1998**	**Lowering of pier, 95 mm**	**D13**	**Sep, 7 1998**	**Rupture of 2 out of 16 tendons**
D05	Aug, 19 1998	Lifting of pier, tilt of foundation	**D14**	**Sep, 8 1998**	**Rupture of 4 out of 16 tendons**
D06	Aug, 20 1998	New reference condition	**D15**	**Sep, 9 1998**	**Rupture of 6 out of 16 tendons**

Training

The undamaged periods used for the training phase were selected based on temperature variations. To properly capture the correlation between temperature and the principal components, both time windows characterized by different absolute temperature levels and those exhibiting significant temperature changes were considered. Figure 2 illustrates the temperature evolution during the monitoring campaign, with the selected training windows highlighted by colored rectangles. The yellow, blue, and red rectangles correspond to periods of nearly stable temperatures at approximately 4, $\breve{}4$, and $20\,^{\circ}C$, respectively, whereas the green rectangle denotes a single interval with pronounced temperature variations.

Once the training data were defined, two different strategies were adopted for the subsequent analysis. The first strategy consisted of selecting, from the first PCA, only those components contributing less than 0.5% of the total variance. The second strategy, which explicitly exploits the temperature information, involved discarding all components whose correlation coefficient with temperature exceeded 0.2, measured through the non-linear Spearman correlation coefficient. The rationale behind these two strategies, both aimed at mitigating the effect of temperature on the DSFs, is illustrated in Figure 3. In particular, Figure 3a reports the bar plot of the temperature-correlation coefficients across the components, highlighting the discarded ones (i.e., those above the threshold) and showing the effects of the correlation-driven selection. The insets compare one of the rejected components with one of the retained ones, clearly revealing the non-linear dependence of the former on environmental variability, which is no longer appreciable in the latter. The idea behind this selection is to discard,

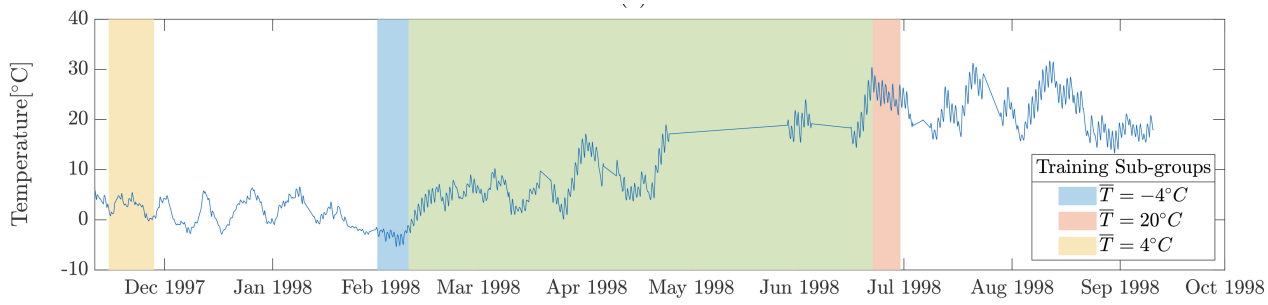


Fig. 2 The temperature during the monitoring period. The colored rectangles show the periods considered in the training phase

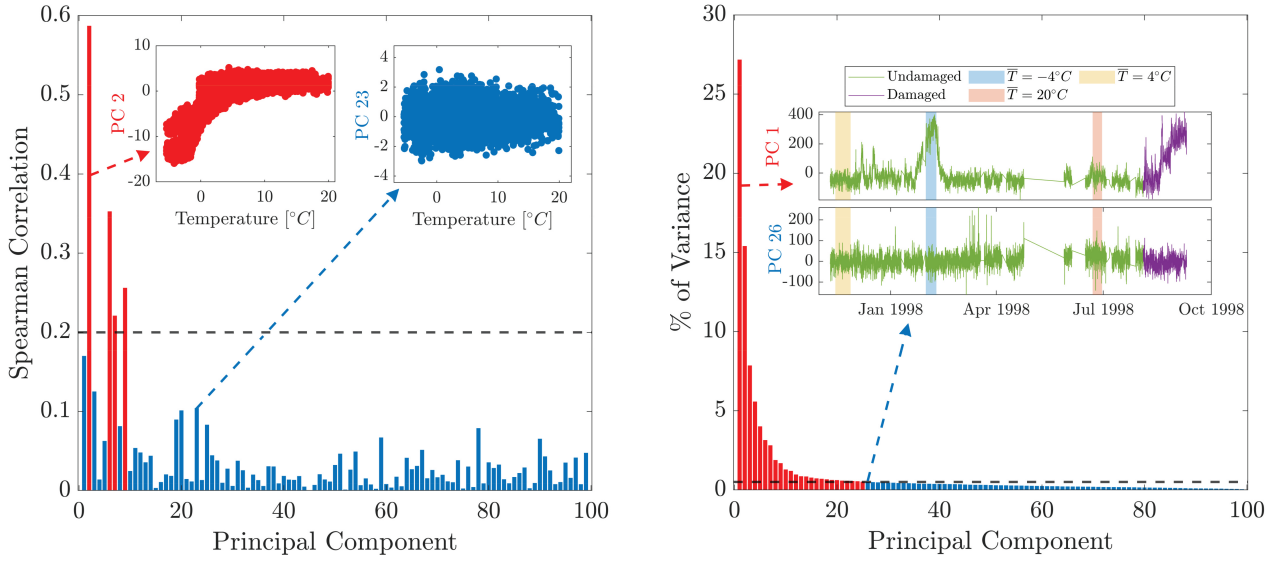


Fig. 3 (a) Spearman correlation between each principal component and temperature. Insets show the projections of the cepstral coefficients onto two selected components: PC 2, which exhibits strong nonlinear dependence on temperature, and PC 23, which is nearly uncorrelated. (b) Percentage of variance explained by each principal component. Insets illustrate the temporal evolution of two projections: PC 1, which mainly reflects environmental variability and shows marked fluctuations at low temperatures (blue window), and PC 26, which is largely insensitive to temperature changes

in a selective manner, the components most affected by temperature, thereby mitigating environmental sensitivity while preserving the overall information.

Similarly, Figure 3b shows the selection of components based on the variance-driven threshold. In this case, the effect is illustrated by comparing the temporal evolution of the first principal component with that of the first minor component. The discarded component strongly reflects the influence of temperature, exhibiting significant deviations during the low-temperature period (blue window). By contrast, the projection onto the first retained PC (i.e., the 26th component) appears insensitive to environmental variability. However, it is also evident that, compared to the previous method, this approach leads to a greater loss of information, and as a result, the deflections associated with damage are mitigated as well, as shown in the insets.

Therefore, the main advantage of the temperature-driven strategy lies in its ability to selectively choose the components to retain. This makes it possible to preserve some of the leading components characterized by large variance (i.e., high information content) but low temperature correlation, thereby constructing a reduced subspace that is insensitive to environmental effects yet highly sensitive to structural damage.

Testing

For the testing phase, the two approaches described during training, one exploiting temperature information and the other based solely on the minor components, were compared using two sets of testing conditions. Undamaged data from the three temperature subgroups, not used during the training phase, were also included to assess the algorithm's sensitivity to EOV effects.

The first set (Case 1) considered two specific damage classes: D04 and D15. These scenarios were chosen because they represent, respectively, the most severe reversible and irreversible damage cases, thus providing a suitable benchmark to preliminarily assess the discriminative capability of the algorithm before performing more refined analyses.

The second set (Case 2) was designed to be more challenging for the algorithm. In this case, the undamaged testing data were considered alongside grouped damage classes: D03 and D04 were merged into a single class, while D13, D14, and D15 formed a second class. The reason behind this grouping is that the structural differences among the damages within each group are minimal, making it unnecessary to distinguish them individually. Consequently, when data from D03 are tested, the algorithm is expected to create a new class and subsequently associate the D04 data with it. Similarly, when analyzing data from D13, the algorithm should define a new class and recognize subsequent damages D14 and D15 as part of the same class.

Results

Different combinations of the parameters of interest (N_{PC2} and $N_{NewClass}$) were investigated to assess the algorithm's performance. Each combination was tested over 10 repetitions to obtain a more reliable indicator. In the plots, the color scale represents the mean value of the binary or multiclass index across the 10 repetitions, while the contour lines indicate the standard deviation, thus providing insight into the variability of the results. The x-axis corresponds to the number of components retained in the second PCA, while the y-axis represents $k = N_{NewClass} - N_{PC2}$. This definition ensures that the number of observations remains larger than the number of features (i.e., N_{PC2}), which is a strict requirement for performing PCA.

Case 1.

Figure 4 reports the results obtained when only damage classes D04 and D15 were considered in the testing phase. The first row shows the performance with the binary index, whereas the second row refers to the multiclass index. For the binary case, no substantial differences are observed, confirming that both approaches successfully distinguish between damaged and undamaged data. In addition, stable performance ($< 94\%$) is achieved across all parameter combinations, with very low variance across repetitions, indicating both robustness and reliability of the results. In contrast, major differences between the two approaches arise for the multiclass index. When temperature information is used to guide the selection of the principal components, better performance is achieved: the index exceeds 0.5 for several parameter combinations, reaching a maximum value of 73%,indicating that the algorithm correctly identifies the three bridge conditions. Conversely, when the variance criterion is adopted, the index remains below 0.5 with low variability for a wide range of parameters (see the blue area in the bottom-right plot of Figure 4), with a maximum value of 68%, suggesting that the algorithm systematically fails to recognize one of the damage classes. This effect is explained by the term s in equation 3, which halves the index value in the case of a missed class. Only for a few parameter combinations the multiclass index reaches values comparable to those obtained when temperature is considered.

To gain further insight into the results, Figure 5 illustrates the evolution of the algorithm, datapoint by datapoint, during the testing phase. The plots were obtained using the temperature-driven strategy, although analogous results are observed when the variance-based strategy is adopted. In particular, cases where the performance index drops below 0.5—such as those forming the blue plateau in Figure 4—correspond to parameter combinations for which the algorithm fails to reliably distinguish class D15 from class D04 (Figure 5a). Conversely, high performance index values indicate a successful separation of the three structural conditions, namely the undamaged state, damage class D04, and damage class D15 (Figure 5b).

A key aspect emerging from the plots concerns the role of *outliers* and *true outliers*. The algorithm classifies as *outliers* all datapoints that cannot be assigned to any class, i.e., those lying outside the defined thresholds, negatively affecting the performance index. However, in experimental datasets, sensor reliability and measurement accuracy must be carefully considered. In the Z24 bridge case, several records are affected by measurement noise or acquisition errors, likely due to malfunctions of the data acquisition system. Such anomalies produce datapoints that markedly deviate from other observations of the same class. These datapoints were identified as *true outliers* when deviating more than three median absolute deviations from the class median. Treating these samples as misclassifications would distort the evaluation of the algorithm.

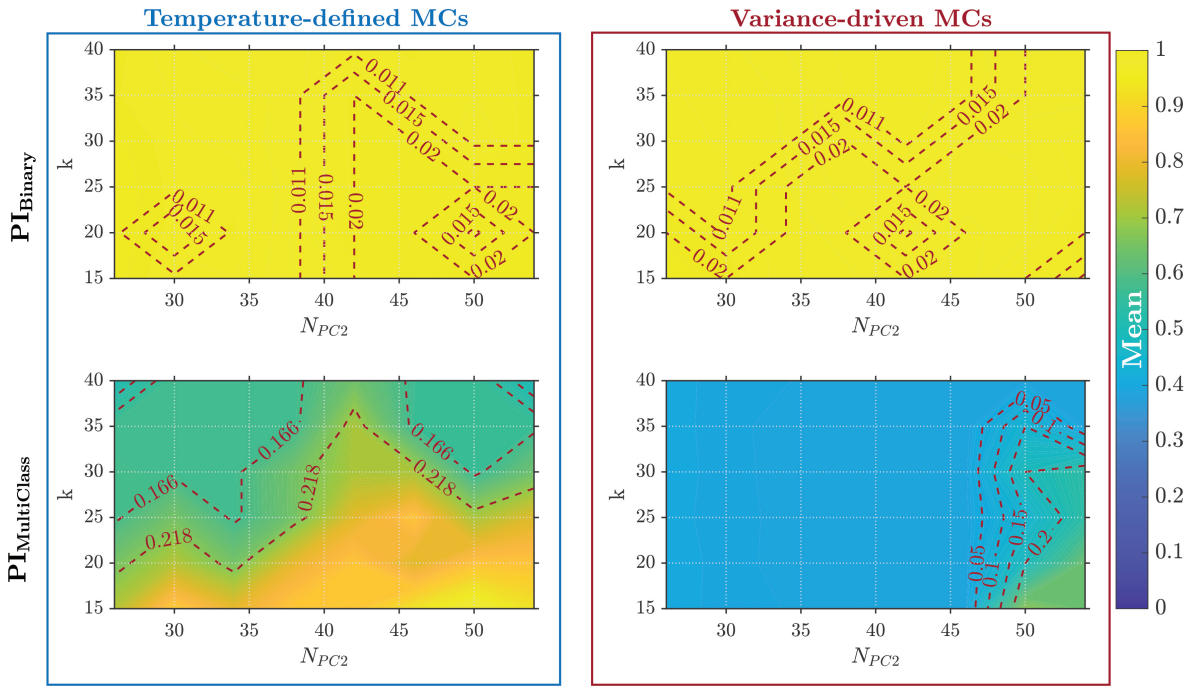


Fig. 4 Contour plots of the mean (color scale) and standard deviation (contour lines) of the performance indices obtained over 10 repetitions for different combinations of N_{PC2} and k in Case 1 (damage classes D04 and D15). The first row refers to the binary index (PI_{Binary}), while the second row refers to the multiclass index ($PI_{MultiClass}$). The left column corresponds to the case where temperature is used as input for the selection of the components, whereas the right column corresponds to the variance-based criterion

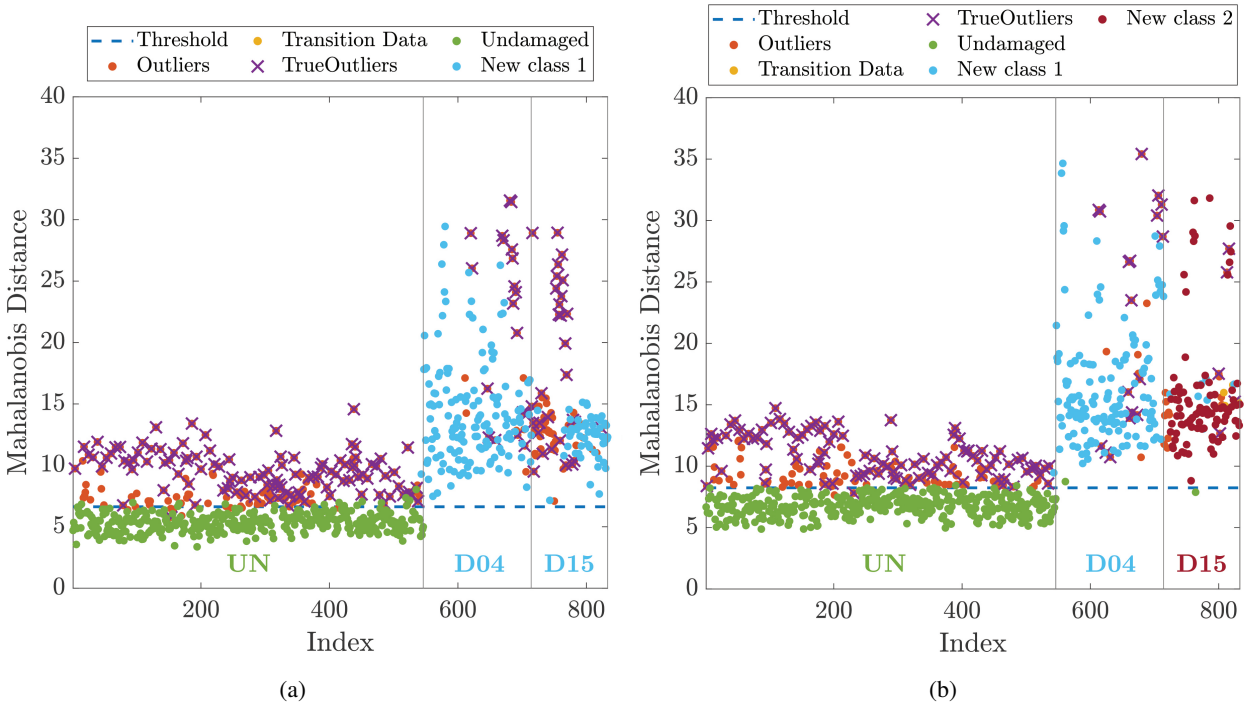


Fig. 5 Squared Mahalanobis distance of points to the undamaged class and their classification for temperature-driven strategy. Results from a single repetition for (a) worst case $N_{PC2} = 30$, $k = 35$; (b) best case $N_{PC2} = 50$, $k = 15$

Therefore, they were included among the *Correctly Classified Data*. Their explicit representation in the plots provides the reader with an indication of their influence on the overall results.

Case 2.

Figure 6 shows the performance indices obtained when classes D03, D04, D13, D14, and D15 are considered in the testing phase, in addition to the undamaged data. It is important to recall that the algorithm is not required to distinguish every individual class, but rather to separate two macro-classes: one grouping data from D03 and D04, and the other grouping the remaining damage classes.

As in Case 1, the binary performance index demonstrates consistently good results ($< 98\%$) across all parameter combinations and repetitions (low variance), both when temperature information is used and when the variance-driven strategy is adopted. In contrast, for the multiclass index, the performance gap between the temperature-based and variance-based approaches is more pronounced than in Case 1. When temperature is employed, good and homogeneous results are obtained across all parameter combinations (maximum 88%). Conversely, the variance-based approach exhibits the same trend already observed in Case 1: a large blue area in the plot indicates low index values with low variance, corresponding to poor performance. Only when a large number of components are retained in the second PCA, combined with small values of k, does the variance-based strategy achieve results comparable to those obtained using the temperature information, with a maximum value of 73%.

To further investigate these outcomes, Figure 7 illustrates examples of both successful and unsuccessful classifications. As in the previous case, the plots are obtained with the temperature-driven strategy, but analogous results hold when the variance-based approach is used. In the worst case (Figure 7a), when the algorithm processes data from class D03, it creates a new class and assigns data from class D04 to it as expected. However, when data from class D13 are analyzed, instead of creating a new class, the algorithm assigns them to the previously created class. The same misclassification occurs for classes D14 and D15. The failure to create a new class halves the multiclass performance index. On the other hand, in the best case (Figure 7b), the algorithm correctly distinguishes between the two macro-classes, namely one comprising classes D03 and D04 and the other including classes D13, D14, and D15, leading to a high performance index.

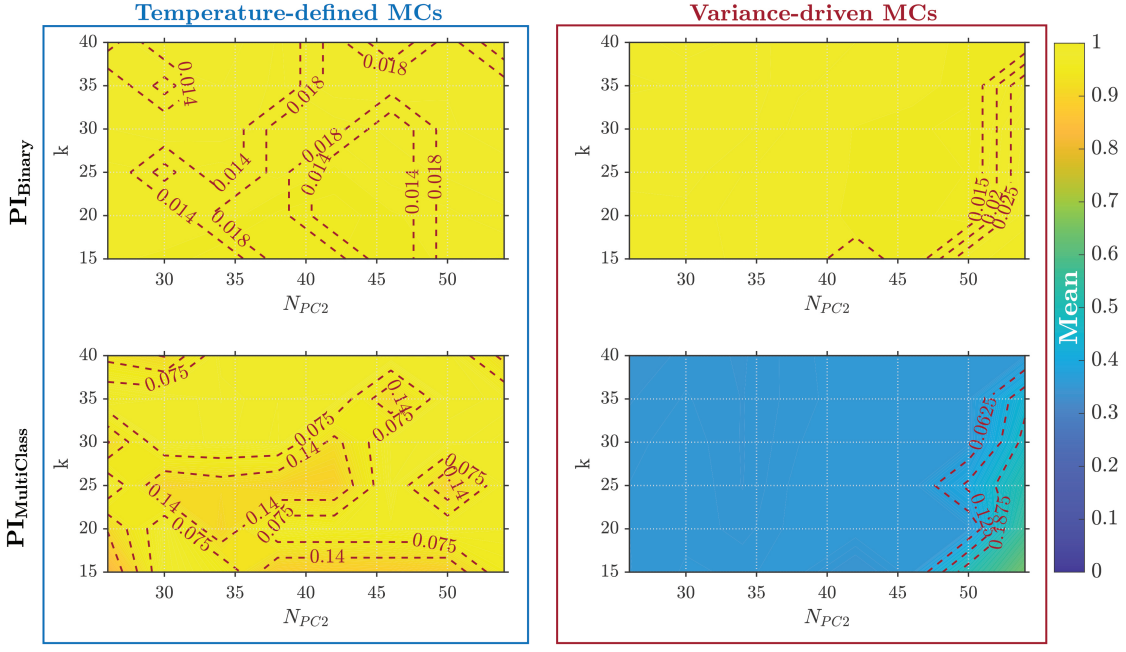


Fig. 6 Contour plots of the mean (color scale) and standard deviation (contour lines) of the performance indices obtained over 10 repetitions for different combinations of N_{PC2} and k in Case 2 (damage classes D03, D04 and D13, D14, and D15). The first row refers to the binary index (PI_{Binary}), while the second row refers to the multiclass index ($PI_{MultiClass}$). The left column corresponds to the case where temperature is used as input for the selection of the components, whereas the right column corresponds to the variance-based criterion

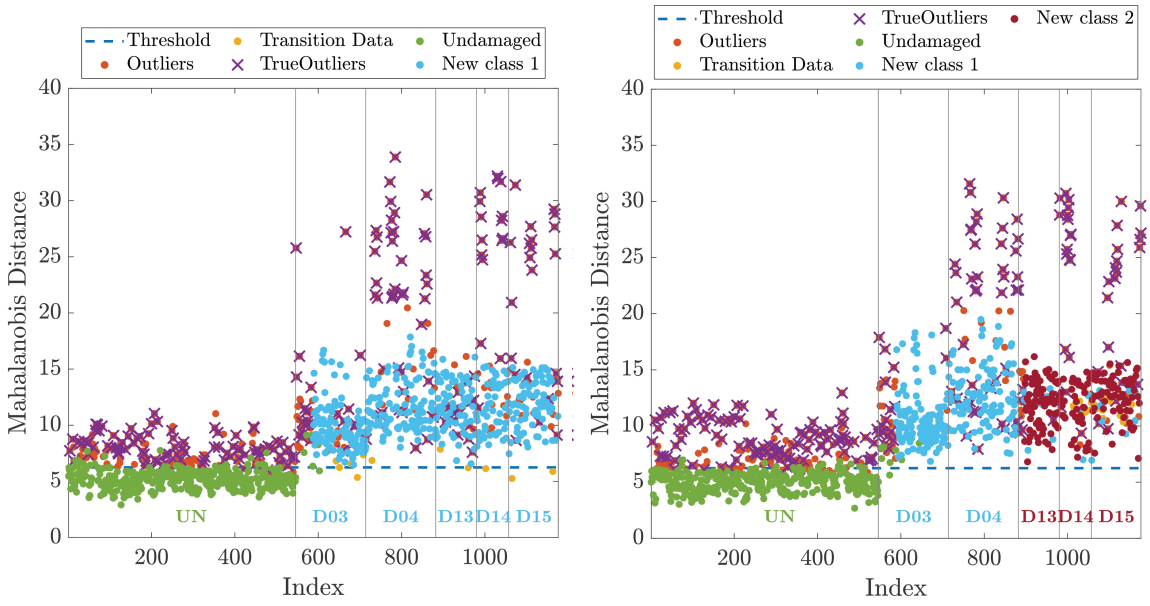


Fig. 7 Squared Mahalanobis distance of points to the undamaged class and their classification for temperature-driven strategy. Results from a single repetition for (a) worst case $N_{PC2} = 26$, $k = 15$; (b) best case $N_{PC2} = 26$, $k = 35$

Conclusions

This study presented an unsupervised methodology for quasi-real-time damage detection and classification based on cepstral features as damage-sensitive indicators. The approach was designed to mitigate environmental influences while enabling the recognition of new damage conditions. Undamaged data were first used to define a reduced-dimensional subspace via Principal Component Analysis. Two strategies for component selection were investigated: a temperature-driven approach, which discards the components most affected by nonlinear correlations with temperature (i.e., explicit approach), and a variance-driven approach, which excludes components carrying more than 0.5% of the variance without using environmental information (i.e., implicit approach). Testing data were then progressively projected into this reduced space, and a second PCA step was employed to enhance separability and support new class creation.

The methodology was validated on experimental data from the monitoring campaign of the Z24 bridge. A multivariate analysis of parameter combinations provided insights into the influence of the number of retained components in the second PCA (N_{PC2}) and the number of data points needed to create new classes.

Results indicate that both strategies achieve near-perfect binary classification between undamaged and damaged conditions. However, when the classification of multiple damage scenarios is required, the temperature-driven strategy clearly outperforms the variance-driven one. In the case of damage classes D04 and D15, the temperature-based approach reached a maximum accuracy of 73%, with robust performance across a broad range of parameter combinations, whereas the variance-based method achieved lower accuracy (68%) and only in a narrow parameter region. The performance gap becomes even more evident when classes D03, D13, and D14 are included: the explicit approach achieved results comparable to binary classification (88%), while the variance-driven approach again provided satisfactory results (65%) only in limited parameter settings.

In summary, the proposed algorithm demonstrates excellent performance when temperature information is exploited during training to guide component selection, highlighting a promising direction toward robust and reliable strategies for structural damage detection and classification.

Acknowledgments The authors sincerely acknowledge the Katholieke Universiteit Leuven, Structural Mechanics Section, for providing the data collected within the SIMCES project.

References

1. Farrar, C.R. and Worden, K. "An introduction to structural health monitoring". *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851):303–315 (2007).
2. García Cava, D., Avendaño-Valencia, L.D., Movsessian, A., Roberts, C., and Tcherniak, D. "On explicit and implicit procedures to mitigate environmental and operational variabilities in data-driven structural health monitoring". In *Structural health monitoring based on data science techniques*, pages 309–330. Springer (2021).
3. Peeters, B. and De Roeck, G. "Reference-based stochastic subspace identification for output-only modal analysis". *Mechanical systems and signal processing*, 13(6):855–878 (1999).
4. Zhang, Y.M., Wang, H., Bai, Y., Mao, J.X., and Xu, Y.C. "Bayesian dynamic regression for reconstructing missing data in structural health monitoring". *Structural Health Monitoring*, 21(5):2097–2115 (2022).
5. Roberts, C., Avendaño-Valencia, L.D., and Cava, D.G. "Robust mitigation of eovs using multivariate nonlinear regression within a vibration-based shm methodology". *Mechanical Systems and Signal Processing*, 208:111028 (2024).
6. Deraemaeker, A., Reynders, E., De Roeck, G., and Kullaa, J. "Vibration-based structural health monitoring using output-only measurements under changing environment". *Mechanical systems and signal processing*, 22(1):34–56 (2008).
7. Deraemaeker, A. and Worden, K. "A comparison of linear approaches to filter out environmental effects in structural health monitoring". *Mechanical systems and signal processing*, 105:1–15 (2018).
8. Cross, E.J., Worden, K., and Chen, Q. "Cointegration: a novel approach for the removal of environmental trends in structural health monitoring data". *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 467(2133):2712–2732 (2011).
9. Tronci, E.M., Betti, R., and De Angelis, M. "Damage detection in a rc-masonry tower equipped with a non-conventional tmd using temperature-independent damage sensitive features". *Developments in the Built Environment*, 15:100170 (2023).
10. Sclafani, L., Stagi, L., Tronci, E.M., Betti, R., and Milana, S. "Real-time unsupervised structural damage detection using cepstral features and principal component analysis". *Structural Health Monitoring*, page 14759217251355946 (2025).
11. Bogert, B.P. "The quefrency alanysis of time series for echoes: Cepstrum, pseudoautocovariance, cross-cepstrum and saphe cracking". In *Proc. Symposium Time Series Analysis, 1963*, pages 209–243 (1963).
12. Tronci, E.M., Beigi, H., Betti, R., and Feng, M.Q. "A damage assessment methodology for structural systems using transfer learning from the audio domain". *Mechanical Systems and Signal Processing*, 195:110286 (2023).
13. Stagi, L., Sclafani, L., Tronci, E.M., Betti, R., Milana, S., Culla, A., Roveri, N., and Carcaterra, A. "Enhancing the damage detection and classification of unknown classes with a hybrid supervised–unsupervised approach". *Infrastructures*, 9(3):40 (2024).
14. Stagi, L., Sclafani, L., Tronci, E.M., Betti, R., Milana, S., Culla, A., Roveri, N., and Carcaterra, A. "An unsupervised damage detection strategy for recognizing unseen health conditions in monitoring bridges". In *International Operational Modal Analysis Conference*, pages 196–207. Springer (2024).
15. Li, L., Morgantini, M., and Betti, R. "Structural damage assessment through a new generalized autoencoder with features in the quefrency domain". *Mechanical Systems and Signal Processing*, 184:109713 (2023).
16. Balsamo, L., Betti, R., and Beigi, H. "A structural health monitoring strategy using cepstral features". *Journal of Sound and Vibration*, 333(19):4526–4542 (2014).
17. M.Morgantini, R.Betti, and L.Balsamo. "Structural damage assessment through features in quefrency domain". *Mechanical Systems and Signal Processing*, 147:107017 (2021).
18. Tronci, E.M., Shid-Moosavi, S., and Speciale, C. "Pattern recognition and damage detection in wind turbine monitoring using cepstral coefficients". In *International Conference on Experimental Vibration Analysis for Civil Engineering Structures*, pages 981–990. Springer (2025).
19. McLachlan, G.J. "Mahalanobis distance". *Resonance*, 4(6):20–26 (1999).
20. Peeters, B. and De Roeck, G. "One-year monitoring of the z24-bridge: environmental effects versus damage events". *Earthquake engineering & structural dynamics*, 30(2):149–171 (2001).
21. Roeck, G.D. "The state-of-the-art of damage detection by vibration monitoring: the simces experience". *Journal of Structural Control*, 10(2):127–134 (2003).
22. Brincker, R., Andersen, P., and Cantieni, R. "Identification and level i damage detection of the z24 highway bridge". *Experimental techniques*, 25:51–57 (2001).

Chapter 10
Leveraging Agentic AI and LLMs to Enhance Conceptual Understanding of Structural Dynamics

Krishna Teja Josyula and Vijaya V. N. Sriram Malladi

Abstract This work introduces a novel educational framework that integrates large language models (LLMs) with agentic AI architectures to help students internalize complex structural dynamics concepts. The focus is on strengthening intuition for how system parameters influence dynamic response in lumped-parameter models. At the core is an interactive environment for single-degree-of-freedom (SDOF) systems, with planned extensions to multi-degree-of-freedom and continuous systems. Specialized AI agents collaboratively guide students through key vibration topics—natural frequency, damping ratio, resonance, transient response, and steady-state forced response—by combining conversational interaction with physics-based reasoning.

The framework employs a LangGraph backend with a NextJS conversational front-end, enabling natural language queries, adaptive explanations, and guided simulation. Integrated computational tools solve governing equations and generate visualizations, creating a seamless transition from conceptual discussion to mathematical solution to animated demonstration. This multimodal design consolidates resources that are traditionally fragmented across textbooks, software, and simulation platforms into a single, unified interface.

The system dynamically adapts to student inquiry, recommending related concepts, visualizations, or simulations aligned with ongoing exploration. For specific parameter configurations or behaviors, it generates contextually relevant video demonstrations that reinforce physical intuition. By unifying conversation, computation, and visualization, the framework supports personalized, adaptive learning and establishes a foundation for AI-driven pedagogical tools across the mechanical engineering curriculum. Potential applications extend beyond undergraduate education into professional training and continuing education.

Keywords Agentic AI · Large language models · Structural dynamics · Adaptive learning

Introduction

Structural dynamics education has long posed significant pedagogical challenges in undergraduate engineering curricula, particularly in conveying the relationships between system parameters and dynamic response behavior in lumped parameter systems. Traditional instruction often struggles to bridge the gap between theory and practice, leaving students grappling with abstract concepts such as resonance [1]. These challenges are compounded by the mathematical rigor required for system analysis, which can create barriers to engagement and limit deep conceptual understanding [2, 3]. Previous studies have shown that interactive and visual approaches are essential to foster conceptual intuition in vibration-related courses [4, 5, 6].

The emergence of Large Language Models (LLMs) represents a paradigm shift in educational technology, offering new opportunities to transform traditional pedagogical approaches [7] . Since the public release of ChatGPT in late 2022, their use in education has gained significant attention across a wide range of disciplines [8]. LLMs demonstrate strong capabilities in

Krishna Teja Josyula and Vijaya V. N. Sriram Malladi
Vibrations, Intelligent Testing, & Active Learning of Structures (ViTALS) Group, Department of Mechanical and Aerospace Engineering, Michigan Technological University, Houghton, MI 49931
e-mail: garuvuanjaneyakrishna@protonmail.com; smalladi@mtu.edu

© The Author(s), under exclusive license to River Publishers 2026

Kristin B. Zimmerman, PhD (eds.), *Data Science in Engineering, 2026, Vol. 9*, Conference Proceedings of the Society for Experimental Mechanics Series, https://doi.org/10.13052/rp-9788743814207

text generation, natural language understanding, and contextual reasoning, making them especially valuable for developing interactive learning environments.

In engineering education, preliminary studies have explored their potential for problem-solving assistance [9], code generation [10], and conceptual explanation [11]. More recent advances show that fine-tuned LLMs with retrieval-augmented generation can deliver course-specific content in mechanical engineering, with applications in finite element methods yielding measurable improvements in student alignment with course materials [12].

Despite these advances, comprehensive frameworks explicitly tailored to technically demanding topics such as structural dynamics remain scarce. Recent surveys emphasize that LLM agents for education are rapidly evolving, but their application has largely focused on general tutoring and domain-agnostic problem solving rather than specialized mechanical engineering contexts [13]. Promising work such as MechAgents demonstrates that multi-agent LLM systems can collaboratively solve mechanics problems by combining planning, coding, and physics reasoning [14], yet these systems remain primarily research prototypes with limited integration into structured curricula. Similarly, studies on agentic workflows argue that decomposing tasks among multiple AI agents can enhance adaptability and pedagogical coherence, but their educational implementation is still in its infancy [15]. In parallel, multimodal foundation models have been identified as having transformative potential in education, though challenges remain in aligning multimodal reasoning with discipline-specific conceptual learning [16]. Recent efforts to incorporate generative AI into engineering education highlight both the promise of enhanced engagement and the current lack of tailored frameworks for advanced topics such as vibrations and structural dynamics [17]. This paper addresses this gap by introducing an AI-driven, agentic framework designed specifically to integrate language, computation, and visualization for structural dynamics education.

The integration of agentic AI architectures with LLMs offers a novel opportunity to address persistent challenges in structural dynamics instruction [18]. Unlike conventional chatbot implementations, agentic systems coordinate specialized AI agents that provide domain-specific expertise, guide adaptive learning paths, and generate contextually relevant explanations. Such architectures enable the creation of sophisticated educational tools that dynamically respond to individual student queries while preserving coherence across multiple learning objectives and conceptual domains [10].

At present, educational resources for structural dynamics are highly fragmented: textbooks provide theoretical explanations, computational tools support equation solving, and visualization software illustrates system behavior. Students must navigate among these disparate platforms, often losing conceptual continuity and struggling to connect theory with application. This fragmentation imposes cognitive overhead, detracts from core learning goals, and limits the effectiveness of traditional instructional approaches.

In this paper, we present a novel educational framework that integrates Large Language Models (LLMs) with agentic AI architectures to create a unified, interactive learning environment for structural dynamics. Our initial focus is on single degree-of-freedom (SDOF) systems, providing a solid foundation in fundamental vibration principles before extending to multi-degree-of-freedom and continuous systems. The framework employs a LangGraph-based backend that coordinates specialized AI agents, each designed to guide students through core topics such as natural frequency, damping ratio, resonance, transient response, and steady-state forced response.

The system combines physics-based reasoning with conversational AI, enabling students to pose open-ended questions in natural language and receive explanations supported by guided simulation and visualization. By consolidating traditionally fragmented resources—textbooks for theory, computational tools for solutions, and separate software for visualization—into a single interface, the framework provides seamless transitions from conceptual discussion to mathematical computation to visual demonstration. The adaptive architecture further responds to individual student queries by recommending related concepts and generating visualizations that illustrate the corresponding physical phenomena.

Beyond tool development, this work establishes a foundation for broader applications of AI-driven pedagogy in the mechanical engineering curriculum [19, 20]. The framework's adaptive nature supports personalized learning paths, allowing students to progress at their own pace while maintaining engagement with increasingly complex material [21]. In doing so, it addresses a critical need for innovative educational technologies that enhance conceptual understanding and prepare students for the evolving landscape of engineering practice.

The figure 1 illustrates the system workflow of the Resonance framework, which adopts a dual-module architecture for structural dynamics education. Both damped and undamped SDOF modules follow identical processing workflows, ensuring consistency while addressing distinct vibration phenomena. The interaction begins with student input through a conversational UI, integrated with AI toolkit components to support natural language communication. Based on intent detection, the system transitions from conversational response to a back-end pipeline that coordinates context analysis, physics reasoning, parameter calculation, and code generation. This modular design allows students to explore either damped or undamped systems through a unified interface while receiving contextually appropriate explanations and dynamic visualizations. Embedded feedback loops further enable iterative refinement of responses, supporting the framework's adaptive learning objectives.

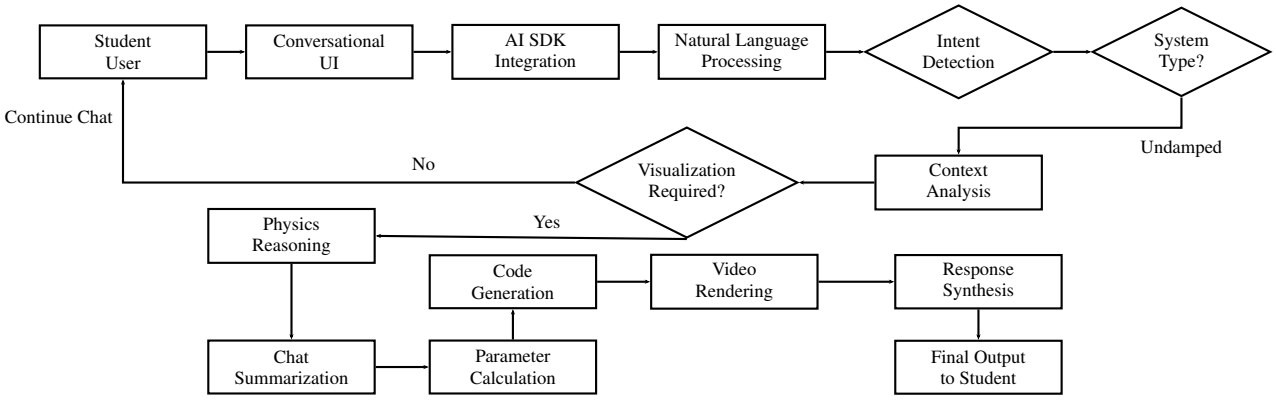


Fig. 1 System Overview: Agentic AI Framework for Structural Dynamics Education demonstrating the seamless integration of damped and undamped system modules within a unified pedagogical platform. The architecture shown represents the undamped SDOF workflow; the damped module follows an identical processing structure with homogeneous architecture.

Methods

This work examines the integration of agentic AI systems in engineering education, particularly in technical domains such as structural dynamics, where traditional pedagogical approaches are limited. To assess the effectiveness of the proposed framework, we pose the following research questions (RQs):

- **RQ1.** Can agentic AI architectures improve undergraduate students' conceptual understanding of structural dynamics compared to traditional instructional methods?
- **RQ2.** How does the integration of conversational AI with in-context reasoning affect student engagement and learning outcomes in mechanical system analysis?
- **RQ3.** What are the pedagogical benefits and limitations of consolidating fragmented educational resources into a unified AI-driven environment?
- **RQ4.** To what extent does the adaptive nature of the framework support personalized learning paths while maintaining rigor in structural dynamics concepts?
- **RQ5.** How does contextual text–video generation enhance conceptual learning when dynamic visualizations are synchronized with conversational explanations?

Each stage of the framework was designed to address specific RQs:

- **Step 1: Conversational Infrastructure.** AI SDK capabilities were embedded directly into the frontend to unify access to resources, addressing RQ3 on the benefits and limitations of consolidating traditionally fragmented tools. Eliminating backend text-generation dependencies ensures a seamless user interface as the foundation for resource integration.

- **Step 2: Pedagogical Intelligence Layer.** Module-specific prompt engineering and a suggestive card interface provide scaffolded learning paths. This phase targets RQ2 (conversational AI's impact on engagement) and RQ4 (supporting adaptive, personalized learning while maintaining rigor). Contextual prompt injection aligns generated explanations with course objectives.

- **Step 3: Specialized Backend Pipeline.** LangGraph orchestration implements sequential nodes for chat summarization, physics reasoning, parameter calculation, and code generation. These enable synchronized text–video responses, directly addressing RQ5 on the role of multimodal visualization in conceptual learning.

- **Step 4: System Integration and Evaluation.** The complete system was deployed and evaluated against traditional instructional methods. This phase confronts RQ1 by testing the overall effectiveness of the agentic AI framework, while also validating RQ5 through real-time multimodal demonstrations of vibration phenomena. Step 4 integrates the complete system architecture and evaluates its effectiveness against traditional methods, directly confronting RQ1's central question of agentic AI effectiveness while simultaneously validating RQ5's text-video synchronization hypothesis through the delivery of synchronized multimodal responses that demonstrate physical phenomena in real-time.

Case Study

This section demonstrates the functional capabilities of the Resonance framework as a proof of concept for agentic AI–driven structural dynamics education. Although the system has not yet been deployed in classroom settings, the case study validates the feasibility of the proposed architecture through systematic testing of its core functionalities: conversational AI interaction, visualization generation, and multimodal response delivery. The demonstration traces the complete workflow—from student query processing through the LangGraph pipeline to synchronized text–video output—providing concrete evidence of the framework's ability to address the research questions outlined above. Representative single degree-of-freedom (SDOF) scenarios are used to evaluate system responses, technical performance metrics, and pedagogical content alignment. Results confirm three key capabilities: (1) consolidation of fragmented educational resources into a unified interface, (2) effective integration of conversational AI for adaptive explanations, and (3) successful implementation of contextual text–video synchronization. Together, these findings establish the foundation for future empirical evaluation with student populations and prepare the framework for expanded educational assessment in subsequent journal studies.

System Architecture Validation

This subsection demonstrates the end-to-end functionality of the Resonance framework. Validation focused on the dual-module design for damped and undamped SDOF systems, confirming that the LangGraph-based backend successfully coordinates with the NextJS frontend via AI SDK integration. Representative test scenarios were implemented to verify correct query routing through the appropriate processing pathways, including context analysis, physics reasoning, parameter calculation, and visualization generation. Figure 2 illustrates the consolidation of damped and undamped system architectures into a cohesive educational platform, empowering students to pursue adaptive learning pathways that integrate theoretical comprehension with visual representation.

Results confirm that both modules exhibit consistent architectural behavior, with identical processing workflows maintained across damped and undamped cases. This validation establishes that the framework's modular design reliably supports multiple system types without requiring architectural modifications, providing a scalable foundation for extension to multi-degree-of-freedom and continuous systems.

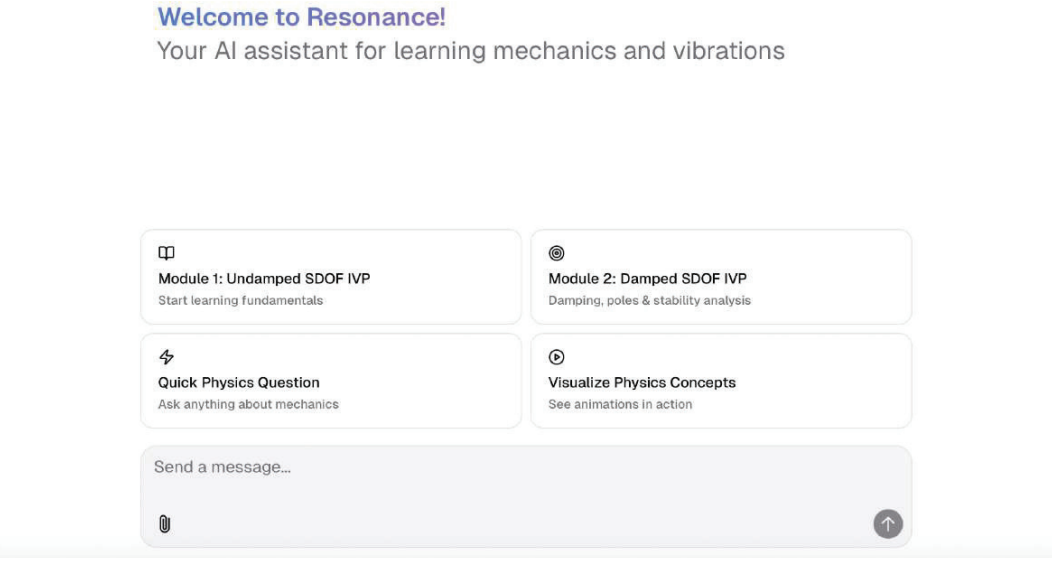


Fig. 2 Dual-module interface showing damped and undamped system selection

Figure 3 illustrates the request–response lifecycle initiated from the front-end as the student interacts with the conversational user-interface. The diagram highlights how user inputs are processed, routed through the AI integration layer, and returned as adaptive responses, demonstrating the seamless flow between the interface and back-end components.

As the student navigates the interface, the LangGraph backend is activated to process each request and generate the corresponding visual explanation of the system under study. Node activations are traced in real time using the LangSmith IDE, a monitoring tool within the LangChain ecosystem, providing transparent tracking of the framework's internal workflows. Figure 4 demonstrates the sequential activation pattern of processing nodes during a typical visualization request, showing the

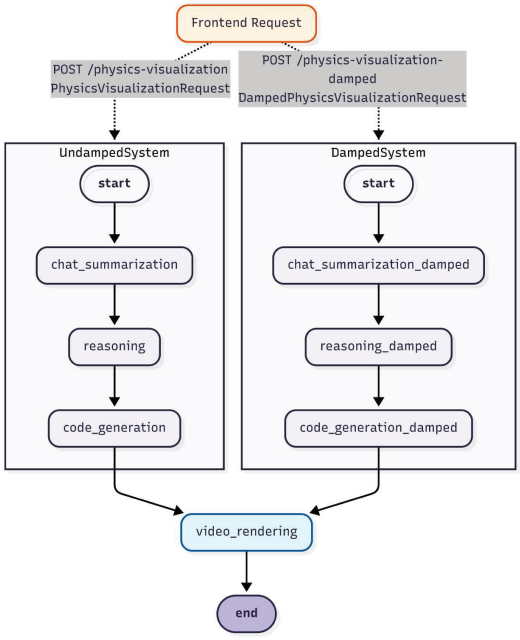


Fig. 3 Request response lifecycle from UI

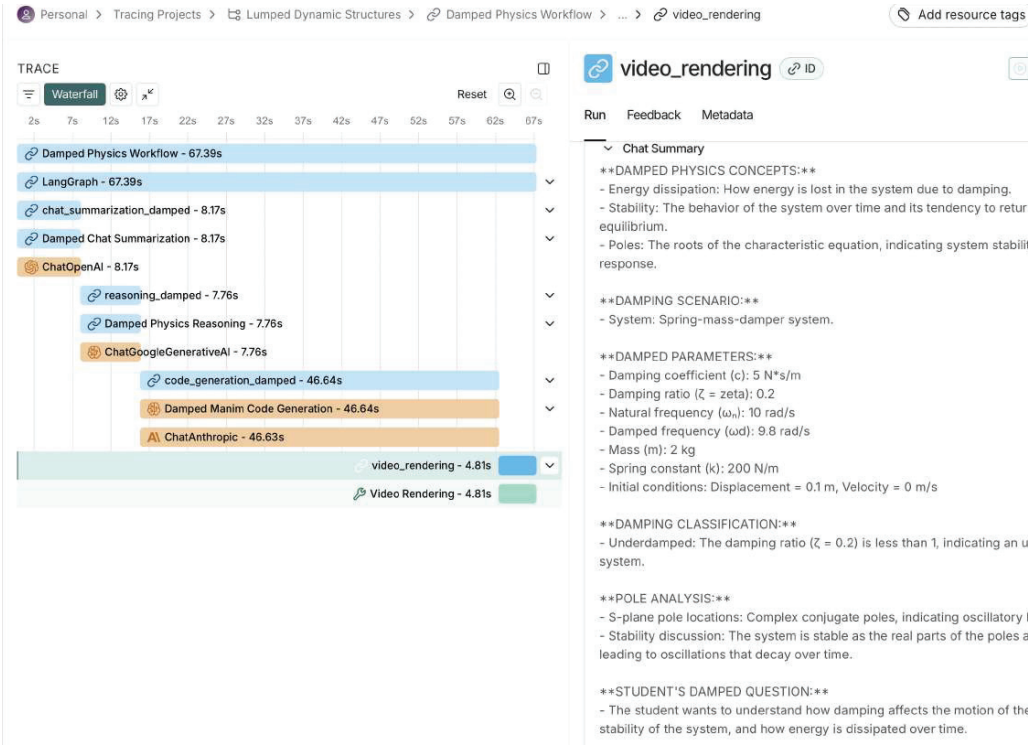


Fig. 4 Real-time tracing of LangGraph node activations during query processing, demonstrating sequential coordination of specialized agents through the backend pipeline

coordination between chat summarization, physics reasoning, parameter calculation, code generation, and video rendering components.

Functional Prototype Demonstration

This subsection presents representative interaction scenarios that highlight the prototype's ability to transition seamlessly from conversational explanations to visualization generation. Example student queries progress from conceptual questions to parameter exploration and ultimately to explicit visualization requests, demonstrating the system's capacity to preserve educational context while adapting to different learning modalities.

Snapshots included below illustrate the complete learning pathway: a student begins with theoretical inquiry, advances through parameter-based analysis, and concludes with synchronized visual explanations. These demonstrations validate the framework's core objective of unifying fragmented resources—conversation, computation, and visualization—into a single adaptive platform.

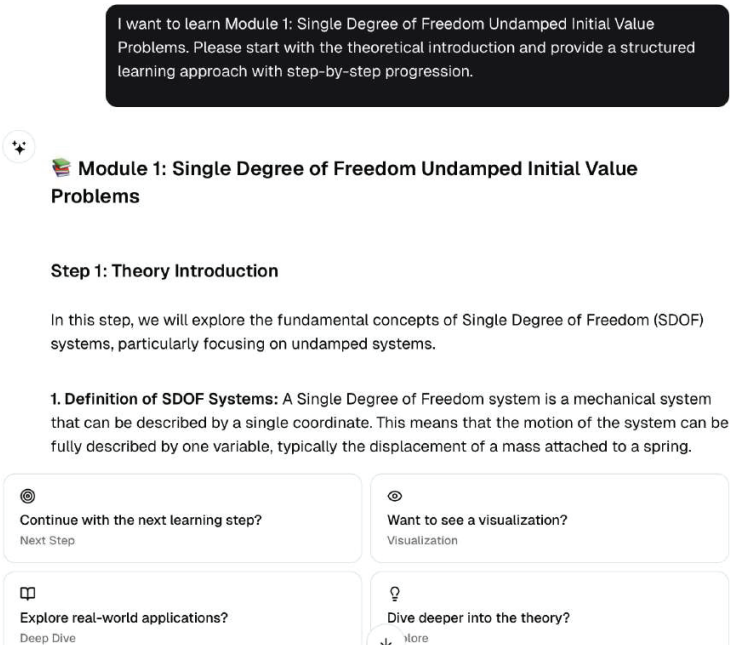


Fig. 5 Initial conversational interaction demonstrating the system's ability to provide conceptual explanations of SDOF fundamentals in response to student inquiry

Conclusion and Future Work

This work addresses a critical gap in structural dynamics education through the development and validation of Resonance, an agentic AI framework that unifies conversational interaction, computational analysis, and dynamic visualization within a single adaptive platform. The framework demonstrates technical feasibility in consolidating resources that have traditionally been fragmented across textbooks, computational tools, and simulation software—addressing persistent challenges in conveying abstract concepts such as resonance, parameter sensitivity, and dynamic response behavior.

The case study validates three foundational capabilities that establish the framework's readiness for empirical evaluation: (1) successful integration of conversational AI with physics-based reasoning to provide contextually appropriate explanations, (2) reliable implementation of dual-module architecture supporting both damped and undamped SDOF systems with homogeneous processing workflows, and (3) effective synchronization of text-based explanations with dynamically generated video demonstrations. The LangGraph-based backend coordinates specialized agents through sequential nodes—context analysis, physics reasoning, parameter calculation, code generation, and video rendering—while the NextJS frontend with AI SDK integration enables natural language interaction and seamless response delivery.

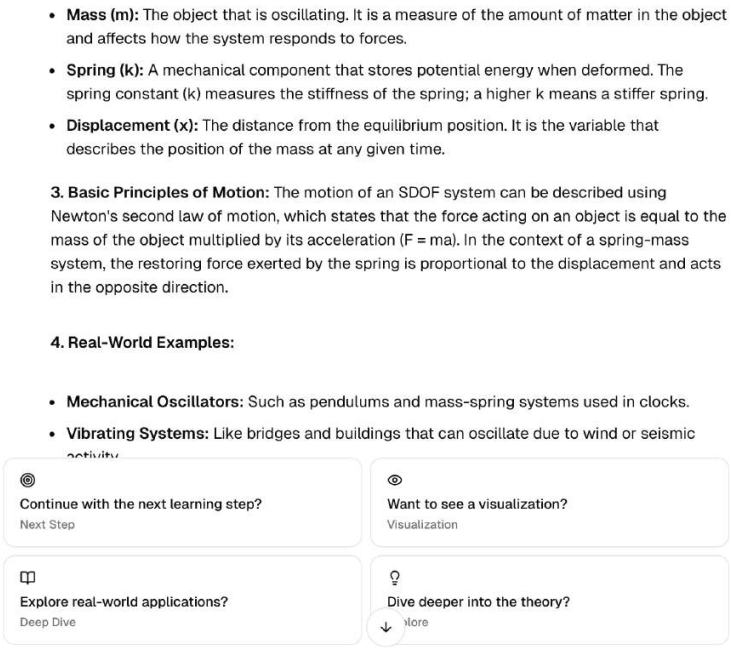


Fig. 6 Integrated learning interface showing the seamless transition from conversational AI guidance to parameter-based exploration

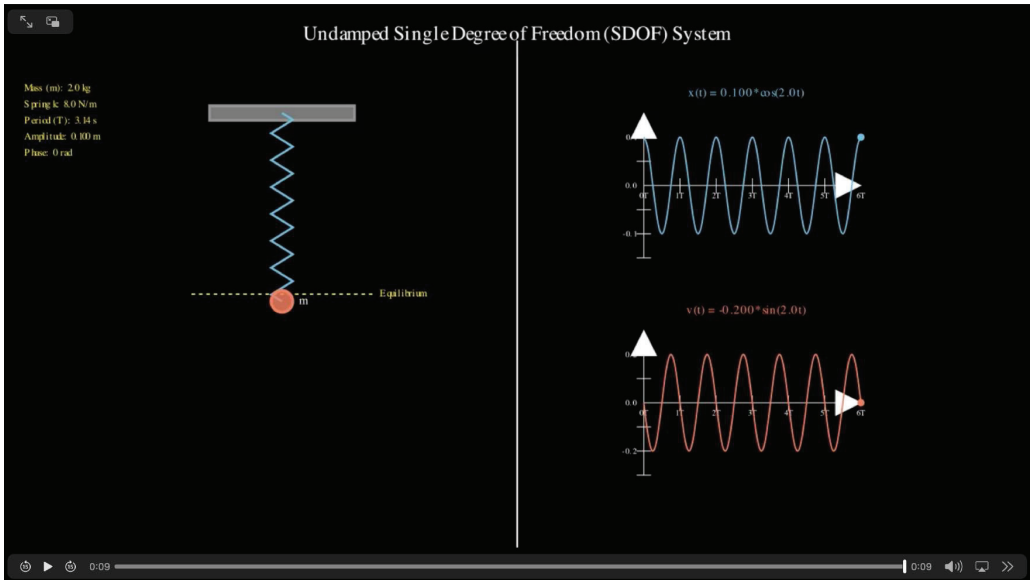


Fig. 7 Real-time undamped SDOF system demonstration with synchronized animation and analytical plots

While the architectural validation confirms the system's technical robustness, this work represents foundational infrastructure rather than conclusive evidence of pedagogical effectiveness. The framework has not yet been deployed in classroom settings, and empirical assessment of learning outcomes remains essential future work. Controlled studies comparing the

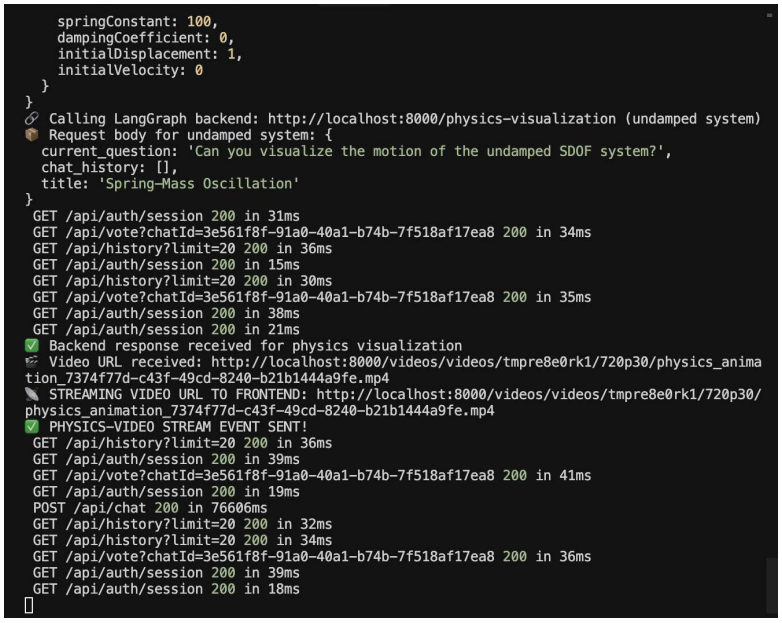


Fig. 8 Streaming transition moment capturing the real-time generation and delivery of multimodal educational content

framework against traditional instructional methods will be necessary to address the research questions posed under "Methods", particularly regarding conceptual understanding improvement (RQ1), student engagement (RQ2), and the pedagogical impact of multimodal text-video synchronization (RQ5).

The modular architecture positions the framework for systematic expansion beyond single degree-of-freedom systems. The consistent processing workflows demonstrated across damped and undamped modules suggest scalability to multi-degree-of-freedom systems and continuous structures without requiring architectural redesign. Furthermore, the agent-based approach enables domain extension—the same coordination patterns between specialized agents could be adapted to other technically demanding subjects in mechanical engineering, including finite element analysis, fluid dynamics, and thermodynamics.

Future work will proceed along three parallel trajectories. First, longitudinal classroom deployment with undergraduate engineering students will assess learning outcomes, engagement metrics, and conceptual retention compared to traditional instruction. Second, the framework will be extended to encompass forced vibration scenarios, multi-degree-of-freedom systems, and continuous systems, expanding coverage across the structural dynamics curriculum. Third, investigation of adaptive learning pathways will examine how the system's recommendation mechanisms influence student exploration patterns and conceptual development over time.

The integration of agentic AI architectures with domain-specific educational content represents a paradigm shift in engineering pedagogy. By demonstrating that conversational AI, computational reasoning, and dynamic visualization can be unified within a coherent framework, this work establishes a foundation for AI-driven educational tools that preserve rigor while adapting to individual student needs. As large language models continue to advance and agentic architectures mature, frameworks such as Resonance offer a pathway toward personalized, adaptive learning environments that bridge the persistent gap between theoretical instruction and conceptual intuition in technically demanding disciplines.

Declaration of generative AI and AI-assisted technologies in the writing process
While preparing this work, the author(s) used Claude to edit the language. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

Acknowledgments This research was partially supported by the National Science Foundation through the grant CMMI-2301776.

References

1. L. D. Feisel, A. J. Rosa, The role of the laboratory in undergraduate engineering education, Journal of engineering Education 94 (1) (2005) 121–130.

2. T. A. Litzinger, P. V. Meter, C. M. Firetto, L. J. Passmore, C. B. Masters, S. R. Turns, G. L. Gray, F. Costanzo, S. E. Zappe, A cognitive study of problem solving in statics, Journal of Engineering Education 99 (4) (2010) 337–353.
3. M. J. Prince, R. M. Felder, Inductive teaching and learning methods: Definitions, comparisons, and research bases, Journal of engineering education 95 (2) (2006) 123–138.
4. Teaching engineering mechanics courses using active engagement methods.
5. S. Di Carlo, L. Benedetti, E. Di Gialleonardo, Teaching by active learning: A laboratory experience on fundamentals of vibrations, International Journal of Mechanical Engineering Education 50 (4) (2022) 869–882. doi:10.1177/03064190211017606.
6. N. Zhu, Embedding active learning and design-based projects in a noise and vibration course for the undergraduate mechanical engineering program, International Journal of Mechanical Engineering Education 50 (1) (2022) 78–88. doi:10.1177/0306419021996151.
7. A. Johri, A. S. Katz, J. Qadir, A. Hingle, Generative artificial intelligence and engineering education, Journal of Engineering Education.
8. S. Filippi, B. Motyl, Large language models (llms) in engineering education: A systematic review and suggestions for practical adoption, Information (2024).
9. M.-L. Tsai, C. W. Ong, C.-L. Chen, Exploring the use of large language models (llms) in chemical engineering education: Building core course problem models with chat-gpt, Education for Chemical Engineers (2023).
10. T. Wang, N. Zhou, Z. Chen, Enhancing computer programming education with llms: A study on effective prompt engineering for python code generation.
11. C. Zhang, Y. F. Zhao, R. El Haddad, Understanding the impact of applying large language model in engineering design education, Journal of Computing and Information Science in Engineering.
12. M. F. Shojaei, R. Gulati, B. A. Jasperson, S. Wang, S. Cimolato, D. Cao, W. Neiswanger, K. Garikipati, Ai-university: An llm-based platform for instructional alignment to scientific classrooms.
13. W. Xu, H. Li, Y. Zhao, et al., Llm agents for education: Advances and applications, arXiv preprint arXiv:2503.11733 (2025). URL https://arxiv.org/abs/2503.11733
14. Y. Wu, Y. Sun, J. Zhang, et al., Mechagents: Large language model multi-agent collaborations can solve mechanics problems, arXiv preprint arXiv:2311.08166 (2023). URL https://arxiv.org/abs/2311.08166
15. Q. Li, R. Chen, A. Gupta, Evolution of ai in education: Agentic workflows, arXiv preprint arXiv:2504.20082 (2025). URL https://arxiv.org/abs/2504.20082
16. A. Gupta, J. Lee, S. Wang, On opportunities and challenges of large multimodal foundation models in education, npj Science of Learning 10 (1) (2025) 12. doi:10.1038/s41539-025-00301-w.
17. Z. Zhang, Y. Chang, Leveraging generative ai to enhance engineering education at both low-level and high-level study, in: 2024 Fall ASEE Mid-Atlantic Section Conference, ASEE, 2024.
18. R. Van Campenhout, D. Soto-Karlin, M. Selinger, B. Jerome, Learning engineering in practice: A case study on developing llm-based educational tools, in: International Conference on Human-Computer Interaction.
19. J. Pereira, J.-M. López, X. Garmendia, M. Azanza, Leveraging open source llms for software engineering education and training, in: 2024 36th International Conference on Software Engineering Education and Training (CSEE&T).
20. V. D. Kirova, C. S. Ku, J. R. Laracy, T. J. Marlowe, Software engineering education must adapt and evolve for an llm environment.
21. M. Bernabei, S. Colabianchi, A. Falegnami, F. Costantino, Students' use of large language models in engineering education: A case study on technology acceptance, perceptions, efficacy, and detection chances, Computers and Education: Artificial Intelligence.
22. M. Z. Li, Using prompt engineering to enhance stem education, in: 2024 IEEE Integrated STEM Education Conference (ISEC).

Chapter 11
AI-Enhanced Anomaly Detection of Acoustic Emission Sensor Data via Clustering and Noise Discrimination: A Machine Learning-Based Study

Ali Bakhtiari, Tonghao Zhang, Antonio De Luca, and Nezar Abraham

Abstract Acoustic Emission (AE) monitoring has emerged as a powerful non-destructive testing (NDT) and structural health monitoring (SHM) technique for identifying early signs of damage in materials and structures. By capturing high-frequency stress waves generated by microcracking, fracture, or material deformation, AE provides unique insight into damage progression. However, AE data are often characterized by high volume, noise contamination, and complex feature distributions, making interpretation challenging. Artificial Intelligence (AI) and machine learning (ML) offer promising pathways for automating AE signal classification, anomaly detection, and real-time decision-making.

Within this study, a cantilever wood strip subjected to an increasing point load at its free end was monitored using AE sensors until failure. A total of 735 AE hits were collected and analyzed through a hybrid framework that combined unsupervised clustering with supervised classification. Five time-domain features were extracted, standardized, and reduced via principal component analysis (PCA) prior to *k*-means clustering. The analysis revealed two distinct clusters, corresponding to fracture-related burst events, and frictional noise from specimen–table interactions alongside minor secondary noise sources. Importantly, the frictional noise identified in this study differs from typical ambient noise, as its signal characteristics increase the likelihood of misclassification in AE-based SHM. Cluster validation using a *k*-nearest neighbor (K-NN) classifier achieved 98.3% accuracy, underscoring the robustness of the proposed methodology. The findings highlight the value of integrating clustering and supervised learning for reliable AE source discrimination. This work contributes to advancing SHM frameworks by demonstrating how challenging noise mechanisms can be identified and separated, thereby improving diagnostic reliability and applicability in real-time monitoring.

Keywords Acoustic emission · Unsupervised learning · Supervised learning · K-means clustering · Noise detection

Introduction

Acoustic Emission (AE) monitoring has become an increasingly important tool in non-destructive testing (NDT) and structural health monitoring (SHM) due to its ability to detect the initiation and progression of damage in real time [1]. AE signals are generated by transient stress waves released during events such as crack initiation, crack growth, fiber breakage, matrix cracking, or other localized deformations within materials [2]. Owing to its sensitivity and real-time capability, AE monitoring has been widely applied in aerospace, civil infrastructure, and mechanical systems to track the health of components and prevent catastrophic failures [3]. However, while AE provides a direct window into damage evolution, the technique is challenged by the complexity of the signals, which are often non-stationary, high-volume, and susceptible to noise contamination [4]. This complexity limits the reliability of conventional threshold-based or parametric interpretation methods, which frequently struggle to distinguish true damage events from irrelevant noise [5].

Ali Bakhtiari · Antonio De Luca
Thornton Tomasetti, Inc., 101 NE 3rd Ave #1170, Fort Lauderdale, FL 33301
e-mail: abakhtiari@thorntontomasetti.com; adeluca@thorntontomasetti.com

Tonghao Zhang · Nezar Abraham
Thornton Tomasetti, Inc., 600 W Fulton St, Suite 705, Chicago, IL 60661
e-mail: txzhang@thorntontomasetti.com; NAbraham@ThorntonTomasetti.com

© The Author(s), under exclusive license to River Publishers 2026

Kristin B. Zimmerman, PhD (eds.), *Data Science in Engineering, 2026, Vol. 9*, Conference Proceedings of the Society for Experimental Mechanics Series, https://doi.org/10.13052/rp-9788743814207

Machine learning (ML) and artificial intelligence (AI) approaches have recently emerged as powerful solutions to address these challenges [6]. By leveraging feature extraction, clustering, and classification techniques, ML offers the capability to identify patterns in large AE datasets that are difficult to detect through traditional methods. Unsupervised learning methods, such as clustering, allow for grouping signals with similar feature characteristics, thereby enabling the identification of underlying physical mechanisms without requiring pre-labeled datasets. Meanwhile, supervised learning algorithms can be used to validate and refine these groupings, providing greater robustness for decision-making in SHM applications [7]. Several studies have successfully employed clustering methods, including *k*-means, hierarchical clustering, and density-based algorithms, to separate AE sources such as matrix cracking, fiber breakage, or delamination in composite materials [8]. Despite these advances, one persistent limitation is the treatment of noise. Ambient noise, operational vibrations, or frictional interactions are often overlooked or treated as background disturbances, even though their signal characteristics can overlap significantly with genuine AE events. Such misclassification can lead to inaccurate assessments of structural condition and false alarms in SHM systems.

The present study addresses this gap by focusing specifically on the identification and separation of friction-induced noise, which differs from typical ambient noise commonly reported in AE literature. The experimental dataset was obtained from a controlled test in which a cantilever wood strip resting on a testing table was subjected to an increasing point load at its free end until failure. During the experiment, 735 AE hits were recorded. These signals were analyzed through a hybrid machine learning framework combining *k*-means clustering with *k*-nearest neighbor (K-NN) classification. Five descriptive features were extracted from each AE hit, normalized, and reduced using principal component analysis (PCA) prior to clustering. The results revealed two distinct clusters, representing fracture-related burst signals, and frictional noise from specimen–table interactions alongside secondary minor noise sources. Notably, the frictional noise identified here exhibited waveforms that differed from ambient background noise but could closely resemble AE signals generated by damage, making it particularly prone to misclassification in both experimental and field monitoring applications. Validation using the supervised K-NN model achieved an accuracy of 98.3%, further demonstrating the robustness of the proposed approach.

This study contributes in two key ways. First, it demonstrates that unsupervised clustering combined with supervised validation provides a reliable means of distinguishing damage-related AE signals from noise, even in relatively small experimental datasets. Second, and more importantly, it highlights the significance of frictional noise as a distinct source of interference in AE monitoring, which has often been overlooked in prior studies. By explicitly identifying and separating this form of noise, the methodology enhances the reliability of AE-based diagnostics and provides a framework that can be extended to broader SHM and NDT applications. The findings underscore the importance of integrating machine learning approaches for anomaly detection in AE monitoring, while also recognizing that not all noise is ambient in origin, some may share signal characteristics with damage and thus demand careful classification.

Experimental Test Setup, Obtained Data, and AE Features

The experimental program was designed to generate acoustic emission (AE) signals during the failure of a wood strip specimen subjected to an increasing point load. The specimen was a rectangular wooden strip measuring 305 mm (12 in) in length, 102 mm (4 in) in width, and 3.2 mm (0.125 in) in thickness. It was positioned on a rigid testing table in a cantilever configuration, with an effective free length of 203 mm (8 in). A monotonic point load was applied at the free end of the strip using a loading fixture, and the magnitude of the load was gradually increased until fracture occurred at the fixed end. This simple test configuration was selected to promote crack initiation and failure under bending while minimizing boundary condition uncertainties.

AE activity was monitored throughout the loading process using a resonant R15α piezoelectric sensor manufactured by MISTRAS Group, Inc. The sensor was mounted on the strip at a distance of 76 mm (3 in) from the fixed end to maximize sensitivity to crack initiation and growth. The sensor was coupled to the specimen surface using a thin layer of silicone grease to ensure efficient acoustic transmission. The signal acquisition system was connected to the sensor via a preamplifier with a fixed gain setting, and waveforms were continuously recorded throughout the experiment. Figure 1 illustrates the experimental configuration, including the specimen geometry, loading setup, and AE acquisition system.

A total of 735 AE hits, or waveforms, were captured during the loading and failure process. Each AE hit corresponds to a transient elastic wave detected by the sensor when a micro-event occurs within the specimen or at the specimen–table interface. From each recorded waveform, five standard AE features were extracted: rise time, hit counts, signal energy, duration, and amplitude. These features were selected due to their proven relevance for AE characterization in prior NDT and SHM studies. Rise time provides information on the signal onset and sharpness, hit counts reflect the number of threshold crossings within a waveform, energy quantifies the cumulative signal strength, duration measures the time span of the signal above threshold, and amplitude represents the maximum signal intensity. Together, these descriptors provide a balanced

characterization of AE signals in both time and intensity domains, forming the input feature set for subsequent machine learning analysis.



(a) (b)

Fig. 1 The experimental setup and data acquisition system: (a) Before applying the increasing point load at the free end, and (b) After breakage/failure of the wood strip specimen

Anomaly/Damage Identification via Machine Learning

The differentiation of acoustic emission (AE) sources that exhibit distinct signal characteristics, such as rise time, frequency centroid, and peak frequency, represents a fundamental step toward identifying the physical mechanisms responsible for material damage. Clustering analysis provides a robust framework for this task by grouping signals with comparable feature attributes, thereby enabling the separation of genuine damage events from spurious sources such as environmental noise or frictional interactions.

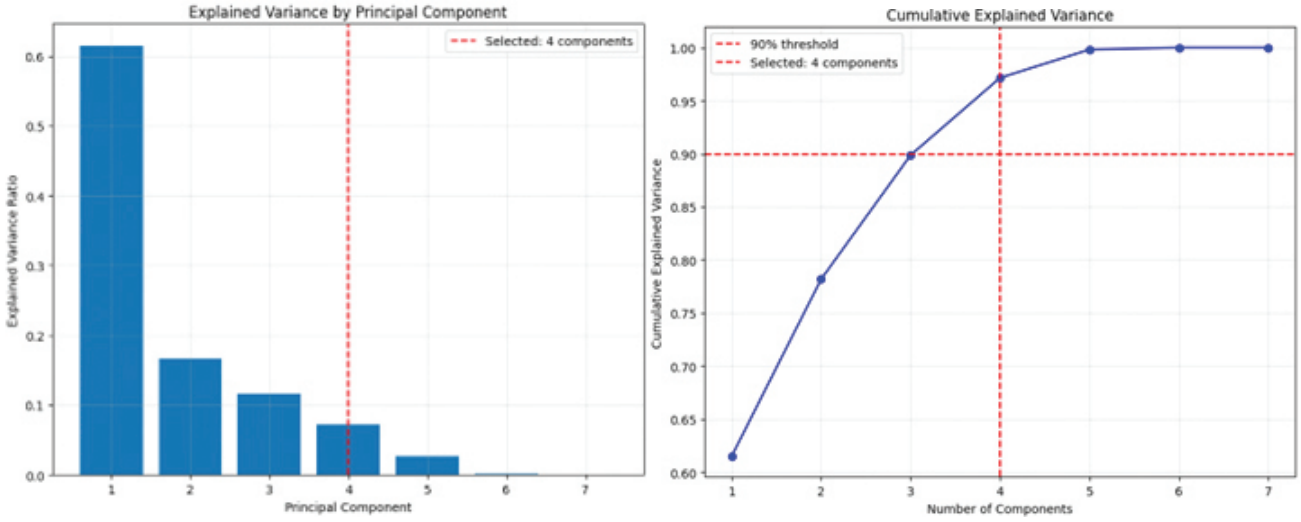
Among the available clustering techniques, the *k*-means algorithm remains one of the most widely employed due to its computational efficiency, scalability, and methodological simplicity. The algorithm partitions a dataset into *k* non-overlapping groups by iteratively assigning each observation to the nearest centroid and subsequently updating centroid positions until convergence is achieved. This iterative process ensures a balance between intra-cluster similarity and inter-cluster separation, making *k*-means particularly suitable for large, high-dimensional AE datasets.

In the present study, *k*-means clustering was implemented to fingerprint AE sources and to classify recorded signals into representative groups associated with two dominant mechanisms: (i) frictional noise caused by specimen–support interactions and (ii) fracture processes leading to wood panel failure. The procedure was first applied to the experimental training dataset to establish baseline AE classes, and the resulting clusters were then validated using a supervised *k*-nearest neighbors (KNN) approach, thereby providing an independent check on classification reliability.

Prior to clustering, the AE signals underwent a structured three-stage preprocessing pipeline. First, five descriptive features, counts, energy, rise time, amplitude, and duration, were extracted for each hit to capture essential time-domain characteristics. Second, all extracted features were standardized to unit variance. This normalization step mitigates disparities arising from heterogeneous physical units and ensures balanced feature contributions during clustering, while also aligning with best practices for preparing data for principal component analysis (PCA). Finally, dimensionality reduction was performed using PCA to facilitate visualization and suppress noise. Four principal components, explaining more than 90% of the cumulative variance, were retained for further analysis, as shown in Figure 2.

The number of clusters was determined using the silhouette coefficient, a well-established metric for evaluating the quality of unsupervised partitions. For each data point, the silhouette value is defined as (b-a)/max (a, b), where a denotes the average intra-cluster distance and b the minimum average distance to the nearest external cluster. The coefficient ranges between −1 and 1, with values closer to 1 indicating strong cohesion within clusters and good separation between clusters. As shown in Figure 3, the silhouette analysis revealed that 2 clusters provided the optimal balance of cohesion and separation, thereby representing the most suitable configuration for the dataset.

While Table 1 summarizes the sample size distribution across the identified clusters, the visualization of the k-means clustering results is presented in Figure 4. The figure displays the AE hits in the reduced principal component (PCA) space, where distinct groupings are clearly observable. The clustering procedure successfully partitioned the dataset into sepa-



(a) (b)

Fig. 2 PCA Analysis: Four principal components explaining more than 90% of the cumulative variance were retained for further analysis

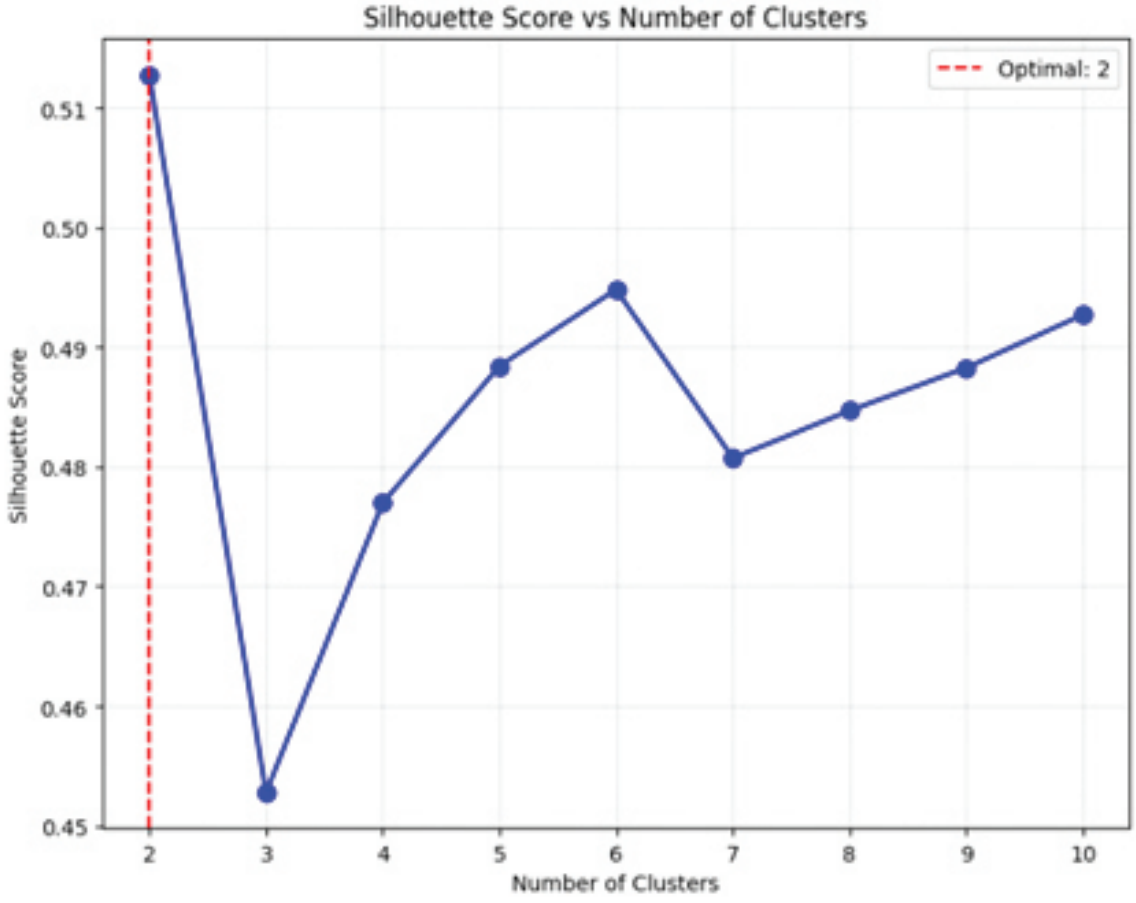


Fig. 3 Optimum number of clusters based on the Silhouette Score

rate clusters, demonstrating its ability to discriminate between signal types based on their extracted features. In particular, two dominant clusters emerged with minimal overlap, indicating that the feature set and dimensionality reduction strategy

Table 1 Number of clusters with their corresponding size distribution and percentage of samples

Cluster	Cluster Size Distribution	Percentage
0	626 Samples	85.2 %
1	109 Samples	14.8 %

provided sufficient discriminatory power to separate AE events associated with different underlying mechanisms. The well-defined grouping observed in Figure 4 confirms the robustness of the unsupervised learning approach in identifying patterns within the AE dataset without requiring prior labeling. This outcome not only validates the suitability of k-means for handling AE signals but also provides the foundation for further interpretation of the clusters in relation to fracture-induced signals and noise-related events. These results confirm that clustering, supported by careful feature engineering and dimensionality reduction, offers a viable and effective means of distinguishing between frictional noise and true fracture events in AE monitoring of wood panel failure.

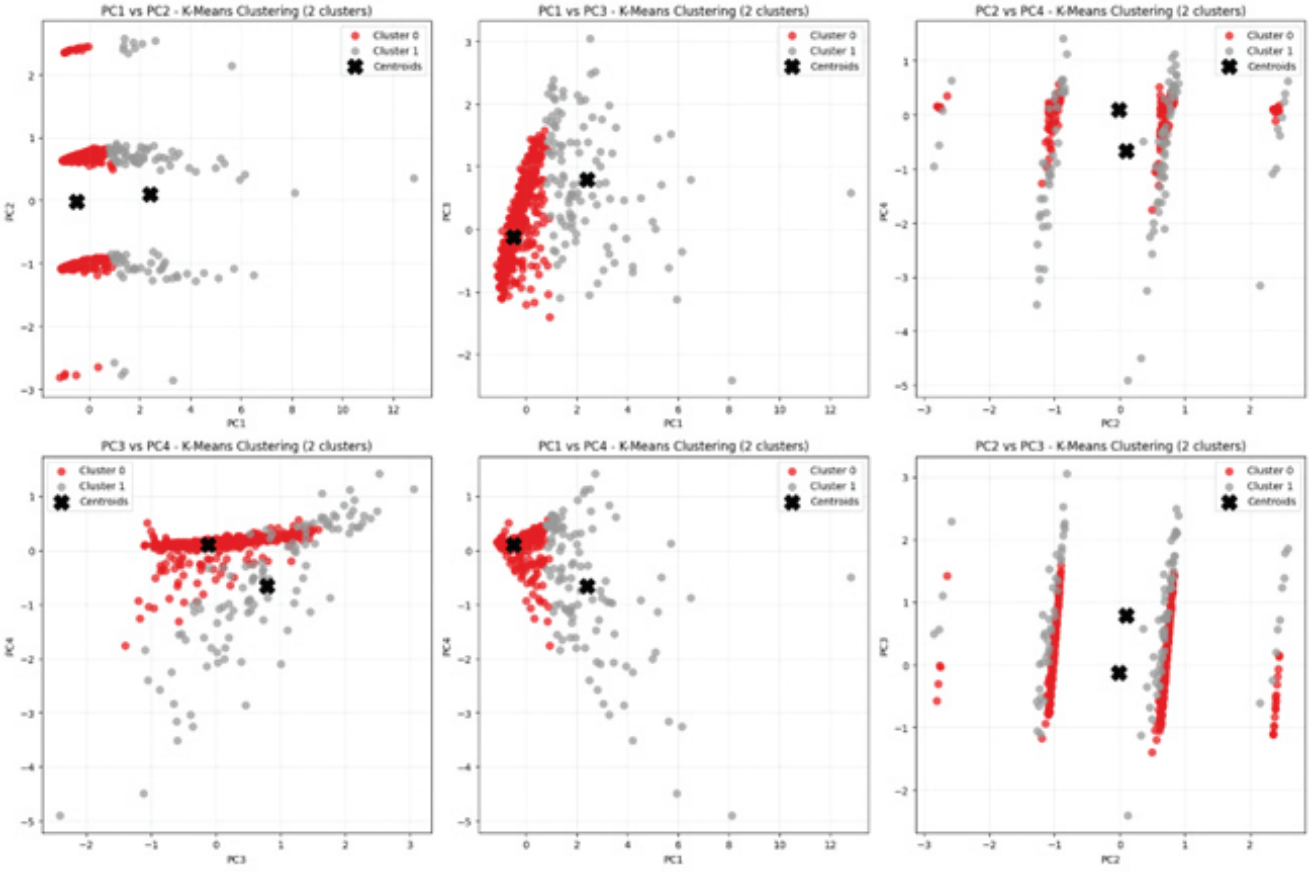
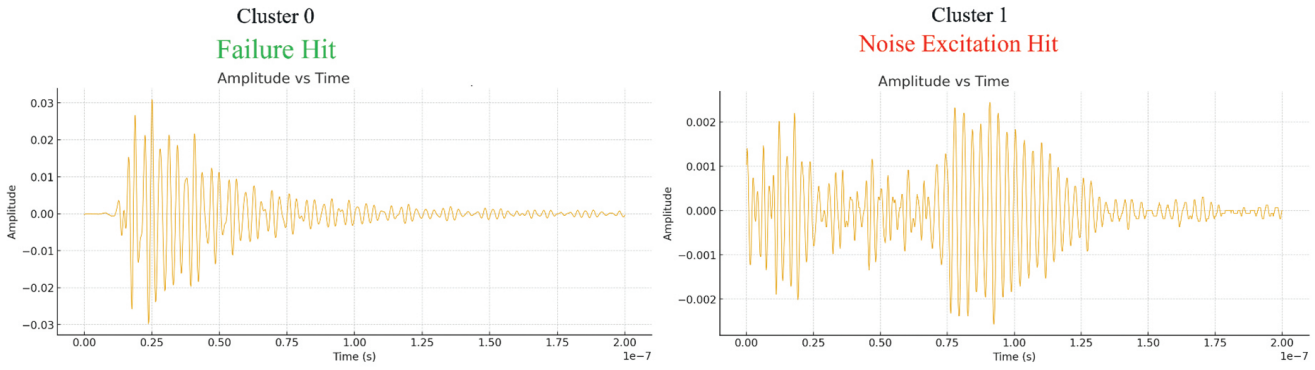


Fig. 4 2-D visualization of the k-means clustering results in the reduced principal component (PCA) space

As illustrated in Figure 5, representative waveforms corresponding to Cluster 0 and Cluster 1 are presented to demonstrate the distinct temporal characteristics of each class. Clear differences in waveform morphology are evident. The waveform associated with Cluster 0, attributed to AE hits generated during the wood strip failure, exhibits a burst-type profile, characterized by a rapid initial rise followed by a sudden attenuation. This sharp, transient form is consistent with the expected signature of fracture-related AE events, where the release of strain energy occurs abruptly as microcracks initiate and propagate.

In contrast, the representative waveform from Cluster 1 displays no such burst-type morphology. Instead, it reveals a more irregular and less impulsive shape, which is indicative of frictional noise. This type of signal is associated with the sliding movement of the specimen across the testing table and the frictional interactions between the two contact surfaces. The



(a) (b)

Fig. 5 Comparison between the two waveforms: Failure Hit vs Noise Excitation Hit

absence of a sharp onset and the presence of extended fluctuations suggest a fundamentally different physical mechanism compared to Cluster 0. Moreover, the maximum amplitude of the AE hit associated with wood strip failure (Cluster 0) was found to be approximately 15 times greater than that of the representative noise waveform (Cluster 1). This pronounced difference in signal intensity provides additional evidence supporting the distinction between the two classes. High-amplitude burst-type signals are generally characteristic of fracture events, where the sudden release of elastic energy produces sharp transients detectable by AE sensors. In contrast, noise-related signals generated by frictional sliding or surface interactions typically exhibit lower amplitudes due to their more gradual and distributed energy release. The observed amplitude disparity therefore, reinforces the interpretation that the clustering algorithm successfully differentiates true failure-related AE events from background noise.

The ability of the clustering procedure to segregate these signal types underscores its effectiveness in distinguishing fracture-induced AE hits from noise-related events. By isolating bursts associated with material failure from frictional artifacts, the method enhances the reliability of AE monitoring for structural diagnostics and provides a more accurate interpretation of the underlying physical processes.

Figure 6 presents the hit rate histograms for all AE signals categorized within each cluster. A clear distinction in event frequency is observed between the two groups. Specifically, the mean hit rate of Cluster 0, corresponding to specimen failure events, is approximately 5.7 times greater than that of Cluster 1, which is associated with frictional noise, sustained over the entire duration of the experiment. This elevated hit rate for Cluster 0 is consistent with the expected increase in AE activity during progressive damage accumulation and eventual structural failure. In contrast, the comparatively lower rate of Cluster 1 indicates that noise-related events occur less frequently and do not dominate the monitoring record. The pronounced difference in event occurrence between the two clusters further validates the ability of the applied clustering framework to isolate genuine fracture-related activity from extraneous noise sources, thereby enhancing the reliability of AE-based diagnostics.

To further evaluate the relationship between the identified clusters and their corresponding AE sources, as well as to assess the robustness of the clustering model, a supervised classification approach was employed using the k-nearest neighbor (K-NN) algorithm. The dataset was partitioned evenly into training and testing subsets, and the dissimilarity between data points was quantified using the Euclidean distance metric. Euclidean distance was selected due to its low computational cost, which is particularly advantageous for real-time monitoring applications where large volumes of AE data are continuously collected. The preprocessing steps, including unit variance scaling and principal component analysis (PCA) for dimensionality reduction, further enhanced the performance of the K-NN model by ensuring balanced feature contributions and mitigating the risk of overfitting in high-dimensional space.

To determine the optimal number of neighbors, multiple values were systematically evaluated. The results indicated that selecting three neighbors yielded the highest classification accuracy, achieving 98.3%. Conversely, increasing the number of neighbors to five reduced performance by introducing misclassifications, while using a single neighbor resulted in susceptibility to outliers and decreased robustness. Therefore, three neighbors were adopted as the optimal configuration for the K-NN model in this study.

The model's predictive accuracy was validated by comparing predicted labels with the actual cluster assignments, as shown in Figure 7. The confusion matrix reveals that the vast majority of AE signals were classified correctly into their respective clusters, with only minimal misclassification observed in off-diagonal entries. These findings confirm that the

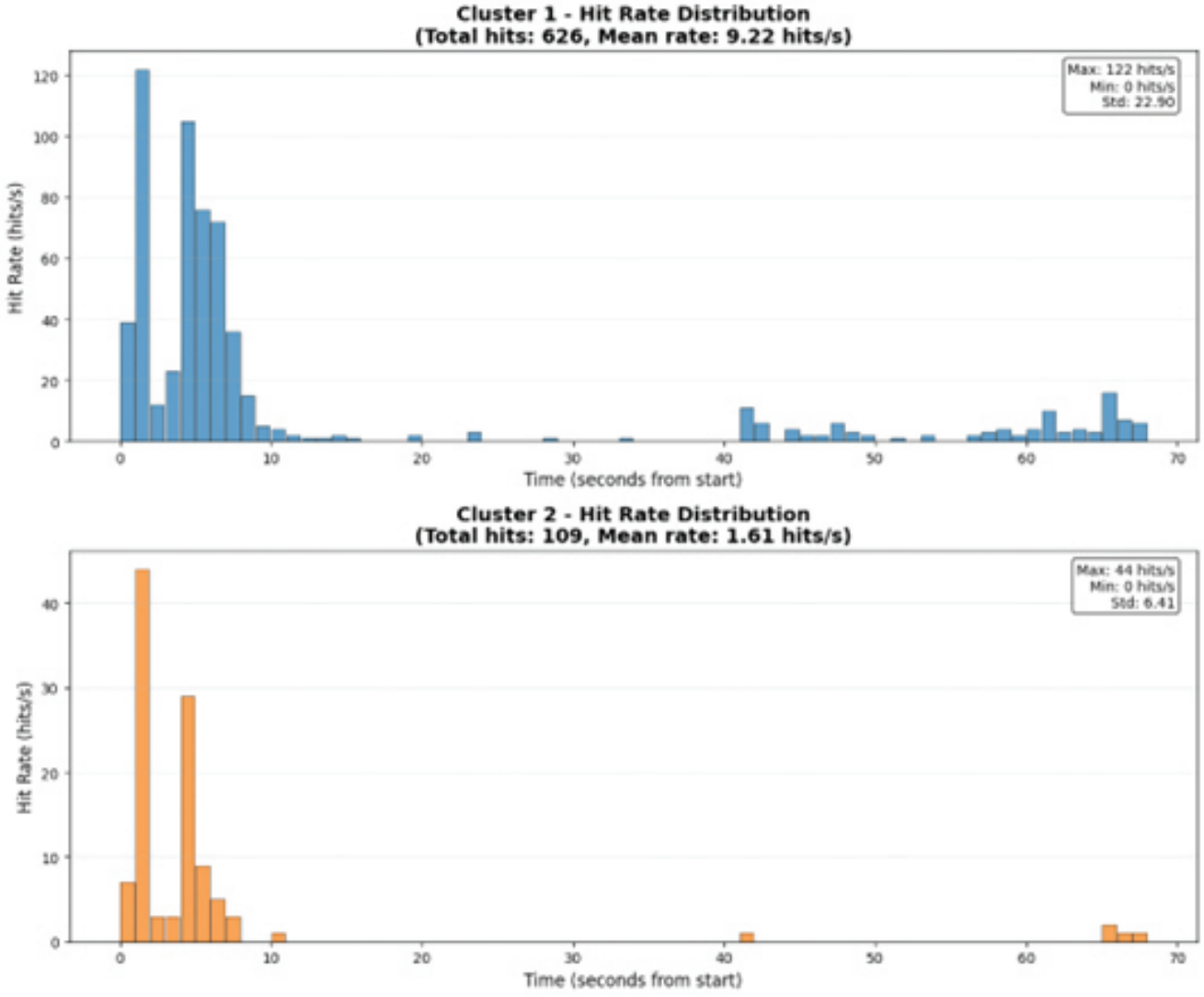


Fig. 6 Hit rate histograms for Cluster 0, corresponding to the specimen's failure, and Cluster 1, associated with frictional noise

supervised K-NN model effectively corroborates the outcomes of the unsupervised clustering, thereby reinforcing confidence in the separation of fracture-related AE events from noise-induced signals.

Figure 8 illustrates the comparison between the true cluster labels of the test set (left) and the predicted labels generated by the K-NN classifier (right). In both plots, the data are visualized in the reduced PCA space, with PC1 and PC2 serving as the horizontal and vertical axes, respectively. The left panel shows the true labels of AE hits in the test set, where three distinct bands corresponding to the identified clusters are visible. These bands reflect the underlying separability of AE signals into meaningful groups associated with fracture-related events and noise sources. The right panel presents the predicted labels obtained using the K-NN model with three neighbors. Correct classifications are denoted by gray points, while misclassified instances are marked with red crosses. The overall prediction accuracy reached 98.3%, with only a small number of data points located near the cluster boundaries being misclassified. This outcome is expected, as boundary regions typically exhibit overlapping feature characteristics, making them more difficult to distinguish with high certainty.

The close correspondence between the predicted and true label distributions demonstrates the robustness of the K-NN model in validating the unsupervised clustering results. The high classification accuracy further supports the conclusion that the applied preprocessing, dimensionality reduction, and optimal neighbor selection collectively contributed to the model's performance. Importantly, the small fraction of misclassified signals does not alter the overall interpretation of the clustering outcome, as the vast majority of AE hits were accurately assigned to their respective groups.

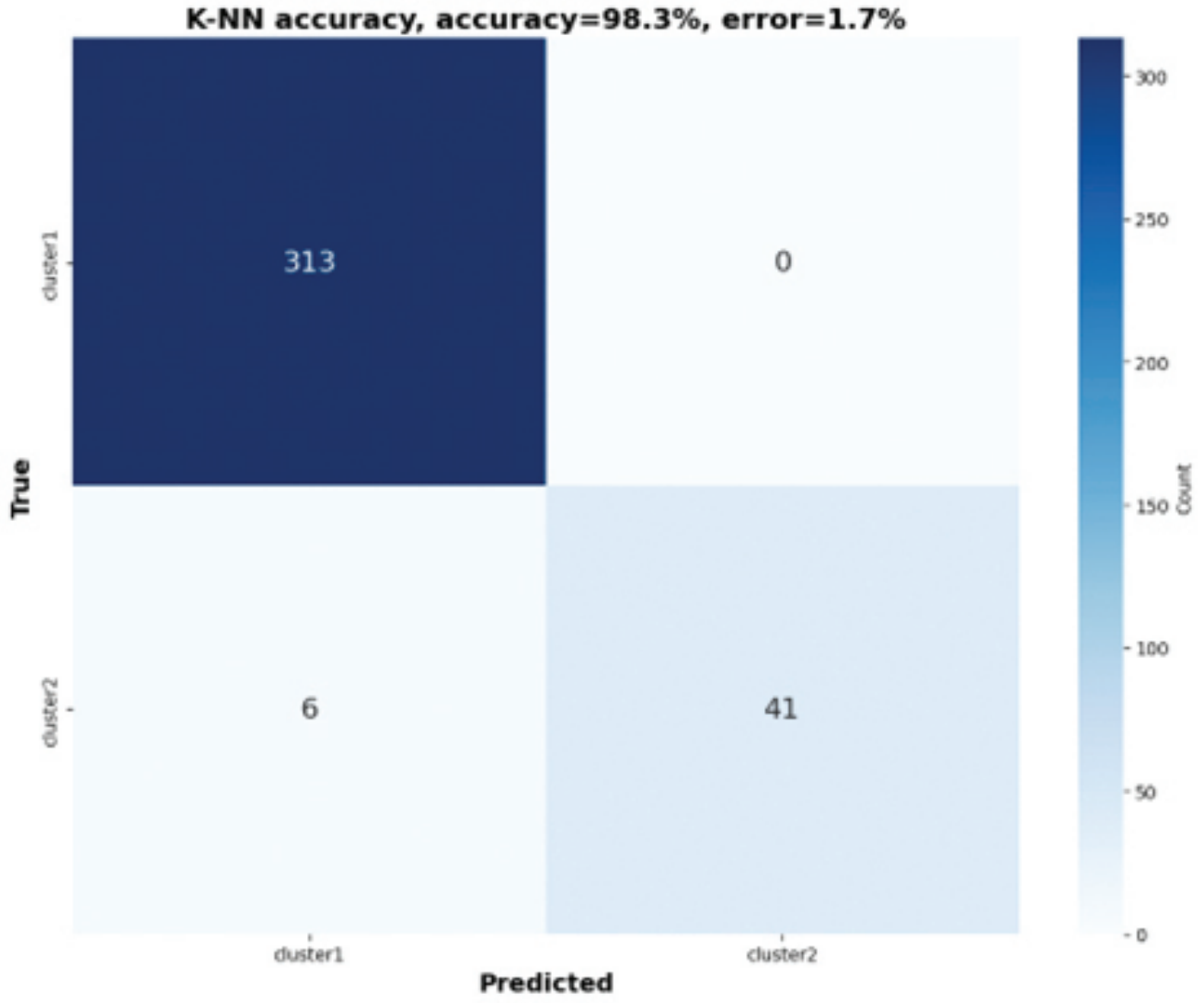


Fig. 7 Confusion matrix for K-NN results showing a prediction accuracy of 98.3 %

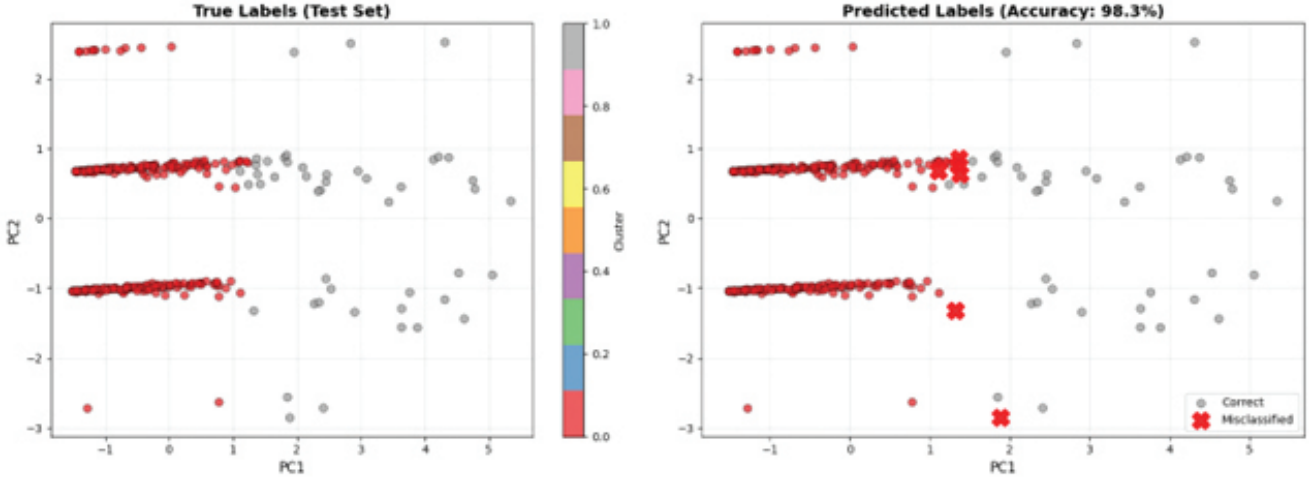


Fig. 8 The comparison between the true cluster labels of the test set (left) and the predicted labels generated by the K-NN classifier (right)

Conclusions

This study investigated the classification of acoustic emission (AE) signals collected during the failure of a cantilever wood strip subjected to an increasing point load at its free end. A total of 735 AE hits were recorded throughout the experiment and systematically analyzed using an unsupervised–supervised hybrid framework combining *k*-means clustering and *k*-nearest neighbor (K-NN) classification. The objective was to distinguish between genuine fracture-induced AE events and noise signals associated with specimen–table interactions. The findings demonstrate that clustering analysis, supported by feature extraction, normalization, and dimensionality reduction, provides an effective means of isolating distinct AE sources. The *k*-means algorithm successfully partitioned the dataset into three clusters, which were subsequently interpreted as (i) fracture-related burst signals, (ii) frictional noise generated by the sliding of the specimen on the table, and potentially minor noise sources. Morphological differences between representative waveforms confirmed the separation, with fracture events exhibiting sharp burst-type profiles and significantly higher amplitudes compared to frictional signals. Quantitatively, the failure-related AE hits showed maximum amplitudes approximately 15 times higher and mean hit rates 5.7 times greater than their noise counterparts, reflecting the dominant role of fracture processes during structural failure. Importantly, the type of noise identified in this study is distinct from typical ambient noise often considered in AE monitoring. Instead of arising from environmental disturbances, the identified noise originated from friction between the specimen and the testing table. This type of frictional noise is particularly challenging, as it shares certain signal characteristics with genuine AE events and therefore has a higher likelihood of being misclassified in similar studies or structural health monitoring (SHM) applications. By explicitly identifying and separating this noise mechanism, the present study contributes to advancing the reliability of AE data interpretation in experimental and field contexts. Validation using the supervised K-NN classifier yielded an accuracy of 98.3%, further confirming the robustness of the clustering approach. Only a limited number of misclassifications were observed, primarily in boundary regions where feature overlap between clusters is expected. Overall, the results confirm that the combination of clustering and supervised validation provides a reliable framework for distinguishing fracture-induced AE activity from noise in experimental testing. Beyond this case study, the proposed methodology has broader applicability to SHM and nondestructive testing, where accurate separation of damage signals from background noise is essential for diagnostic reliability. Future work will focus on extending the approach to larger and more heterogeneous datasets, incorporating additional frequency-domain features, and exploring advanced machine learning classifiers to further enhance real-time AE source discrimination.

Acknowledgments The authors gratefully acknowledge the support of Thornton Tomasetti's Core Lab, whose funding and resources made this study possible. Their commitment to advancing innovation in structural engineering and applied research provided the foundation for the successful completion of this work.

References

1. Melchiorre, Jonathan, et al. "Acoustic emission and artificial intelligence procedure for crack source localization." Sensors 23.2 (2023): 693.
2. Zhang, Tonghao, Chenxi Xu, and Didem Ozevin. "Acoustic emission source localization in complex pipe structure using multi-task deep learning models." Advances in Structural Engineering 28.1 (2025): 23-37.
3. Ahmed, Junaid, et al. "A multiscale indentation-based technique to correlate acoustic emission with deformation mechanisms in complex alloys." *Materials Characterization* 182 (2021): 111575.
4. Tagawa, Y., R. Maskeliunas, and R. Damaševicius. "Acoustic Anomaly Detection of Mechanical Failures in Noisy Real-Life Factory Environments. Electronics 2021, 10, 2329." 2021,
5. König, F., et al. "Machine learning based anomaly detection and classification of acoustic emission events for wear monitoring in sliding bearing systems." *Tribology International* 155 (2021): 106811.
6. Thirumalaiselvi, A., and Saptarshi Sasmal. "Machine learning-based acoustic emission technique for corrosion-induced damage monitoring in reinforced concrete structures." *Engineering Applications of Artificial Intelligence* 137 (2024): 109121.
7. Vos, Kilian, et al. "Vibration-based anomaly detection using LSTM/SVM approaches." *Mechanical Systems and Signal Processing* 169 (2022): 108752.
8. Zhang, Tonghao, et al. "Experimental study on monitoring damage progression of basalt-FRP reinforced concrete slabs using acoustic emission and machine learning." *Sensors* 23.20 (2023): 8356.

Chapter 12
On the Communities of a Population-based SHM Network

Daniel S. Brennan and Keith Worden

Abstract The aspiration of Population-based Structural Health Monitoring (PBSHM) is to accumulate a deeper knowledge of a structure's health, by harvesting available data across a range of similar structures (or substructures). Whilst each structure is represented via a unified language called an Irreducible Element (IE) model, the comparison and similarity metrics are computed within a graph space. As such, each model is subsequently converted into an Attributed Graph (AG) before being placed into the PBSHM Framework's comparison graph space; *the network*.

This paper looks into the application of using the PBSHM Framework's comparison graph space as a complex network and determining if the theories of communities and community structures from network science, can be applied to the PBSHM Framework's network, to determine similarity and define yet-unknown populations.

Keywords Population-based structural health monitoring · Graph neural network · Irreducible element model

Introduction

The intention of Population-based Structural Health Monitoring (PBSHM) [1, 2, 3, 4, 5] is to accumulate additional knowledge on a given structure, by harvesting available data across a range of similar structures — the population. Whilst there are inevitable a myriad of possible methods to establish the aforementioned similarity between real-world structures, this paper will focus on the *Network of Structures* approach defined in the first five papers in the 'Foundations of PBSHM' series [1, 2, 3, 4, 5]. Each structure is represented within the network as a node, weighted links are then established between the structure nodes using the graph-based similarity metrics in the PBSHM Framework.

Previous work in the field of PBSHM has often used a defining feature of the structure as the delimiter for stating what the populations within the network should be — say the number of spans within a bridge. These drawn populations are then used as the judgement for benchmarking if the similarity metric is performing as desired or not. This paper explores the idea of flipping this notion on the head and instead, letting the similarity metrics themselves decide what the populations are within our network via the vehicle of network communities.

The layout of this paper is as follows. The background theory to understanding how structures are represented within the network is set out in the first section. The second section explains how hierarchical clustering and the Ravasz algorithm is used within the PBSHM Network to generate communities — and thus sub-populations — within the network. Lastly a conclusion is given with the next steps to realising this algorithm as a method for generating populations.

Background

Before one can start to compare structures via any form of a similarity metric, one must first have an equal and fair procedure for describing the structures in question — a universal language for capturing the structurally-significant information. The aforementioned vehicle for accomplishing this embedding of information in PBSHM is an Irreducible Element (IE) model [5]. Whilst the scope of the IE-model language is too large to be included within this paper, an abridged recap is included;

Daniel S. Brennan · Keith Worden
Dynamics Research Group, School of Mechanical, Aerospace, and Civil Engineering, University of Sheffield,
Mappin Street, Sheffield, S1 3TD, UK
e-mail: d.s.brennan@sheffield.ac.uk; k.worden@sheffield.ac.uk

© The Author(s), under exclusive license to River Publishers 2026

Kristin B. Zimmerman, PhD (eds.), *Data Science in Engineering, 2026, Vol. 9*, Conference Proceedings
of the Society for Experimental Mechanics Series, https://doi.org/10.13052/rp-9788743814207

however, the interested reader is encouraged to read [5] for a full explanation of the depths of data that can be encapsulated within an IE model.

To generate an IE model of a structure, one must first break down the structure into a set of structurally-significant components — the *elements* of the structure. Once one has a set of elements, these can then be further divided into *regular* elements, which belong to the structure in question, and *ground* elements, which don't belong to the structure, but instead denote the presence of another structure adjacent to the structure one is modelling. One could oversimplify this process to imagine a magic hammer that would instantly reverse any welds, rivets, adhesive, etc that hold a structure together. The set of *element*s would be the components left over after hitting the structure with the aforementioned hammer.

The *relationship*s are the interactions between the components. The welds, rivets, adhesive, etc that the previous metaphor had removed from the structure. Akin to *element*s, the *relationship* objects within the model have four further categorisations. A *perfect* relationship, is when a single larger *element* has been broken down into smaller *element*s for either damage localisation or embedding complex geometries within the model. A *connection* relationship, is where several *element*s are held together by an un-modelled *element* which has been omitted from the model as it has been determined to be structurally non-significant. A *joint* relationship is where the physics between the two *element*s interacting is desired to be modelled. And finally a *boundary* relationship denotes the interaction between a *regular* element and a *ground* element. Each categorisation of both *element*s and *relationship*s has a set of potential information that can be embedded within that type of object in the model.

Figure 1 shows the above process for an example wind turbine. The structure is first broken down into its structurally-significant components: three blades, one rotor hub, one nacelle, three tower parts, and one foundation. These *element*s — and associated *relationship*s — are then converted into a graph domain so as to be compatible with the PBSHM Framework similarity space. It is important to note that the associated graph from an IE model, is not just a purely topological representation, the available model object knowledge is encoded into attributes on the associated vertex or edge.

Once the structure is within the PBSHM Framework's network space, similarity metrics can be computed using the available algorithms in the PBSHM Framework. To date there are four published algorithms that can operate within the PBSHM Network: the Jaccard index with Maximum Common Subgraph (MCS) [2, 6, 7], Graph Kernels [8], Graph Matching Network (GMN) [9], and NetPair [10]. Whilst the middle two algorithms use Machine Learning (ML) [11] to help compute the similarity between two graphs, the first and last algorithms — the Jaccard index with Maximum Common Subgraph and the NetPair algorithm — are graph theory methods [12, 13] for computing similarity.

As the purview of this paper is not to evaluate the available similarity metrics within the framework, the NetPair [10] algorithm has been chosen as the base similarity metric for this paper. Figure 2 shows the standard NetPair similarity metrics against the first five bridges of each span in the PEAR beam-and-slab [14] dataset. As one can see, even though the NetPair algorithm is able to find similarities within the network, the similarity is not confined to the bridges with the same number

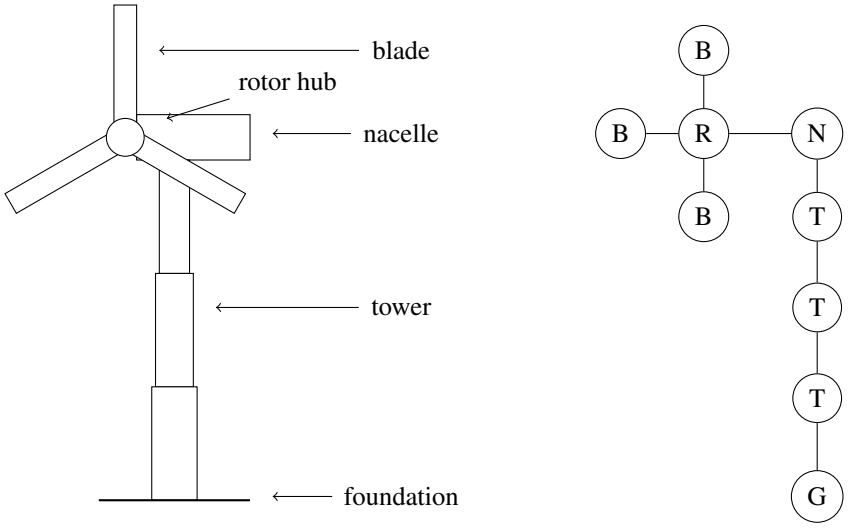


Fig. 1 The process of breaking down a structure (a wind turbine in this example) into its Irreducible Elements and then into its final form as an Attributed Graph (AG) for comparison in the *Network of Structures*

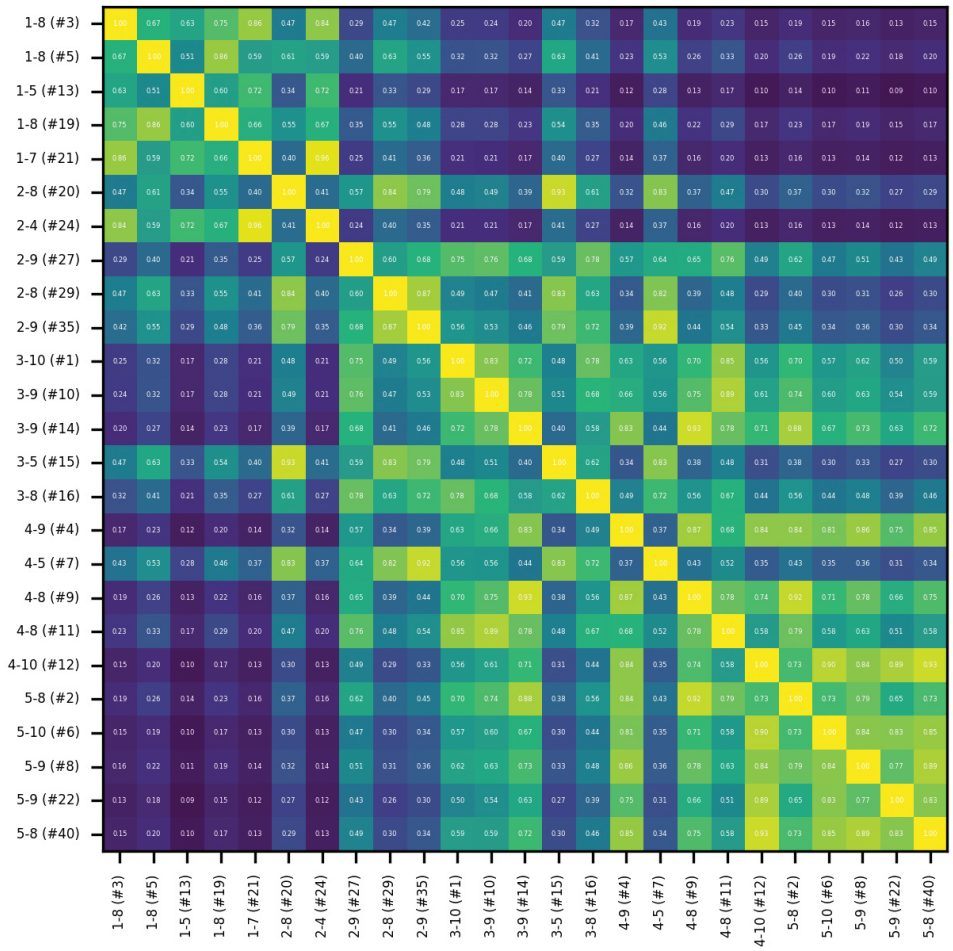


Fig. 2 NetPair similarity heatmap using the first five bridges from each span in the beam-and-slab PEAR dataset

of spans or even neighbouring number of spans. Therefore, one asks; can network communities provide an answer to what populations exist within the network?

Communities

Communities are a method within Network Science [12, 13] of clustering similar nodes together. One of the methods for achieving community discovery is hierarchical clustering. The starting point for hierarchical clustering is a similarity matrix that illustrates the distance between each node within the network. This similarity matrix could be as trivial as calculating the Euclidean distance between the nodes within the network, or it could be as complicated as the output from the PBSHM Framework's similarity comparisons.

Hierarchical clustering can be split into two types of algorithms: *agglomerative* and *divisive*. Agglomerative algorithms merge nodes/communities with a high similarity into a single community, whereas divisive algorithms remove nodes from a community with a low similarity. One example of an agglomerative algorithm is the Razasz algorithm [15]. The Razasz algorithm can be summarised in four steps:

1. Calculate a similarity matrix for the network.
2. Determine a metric for deciding the similarity of a community.
3. Apply hierarchical clustering.
4. Create a dendrogram.

To apply the Razasz algorithm to the PBSHM Network, one first takes the similarity matrix generated by the PBSHM Framework — in the figures included, this is the default NetPair algorithm with all the features weighted equally. Each node — structure — becomes its own community. The similarity matrix is then scanned to find the highest similarity within the network which is not a match to itself — similarity scores from and to the same structure should have the maximum similarity, which is unity. The two associated communities are then merged together and the average of the community member's similarities are stored back within the similarity matrix. The algorithm then goes back to scanning the matrix for the highest similarity score and merging the associated communities until there is only one community left.

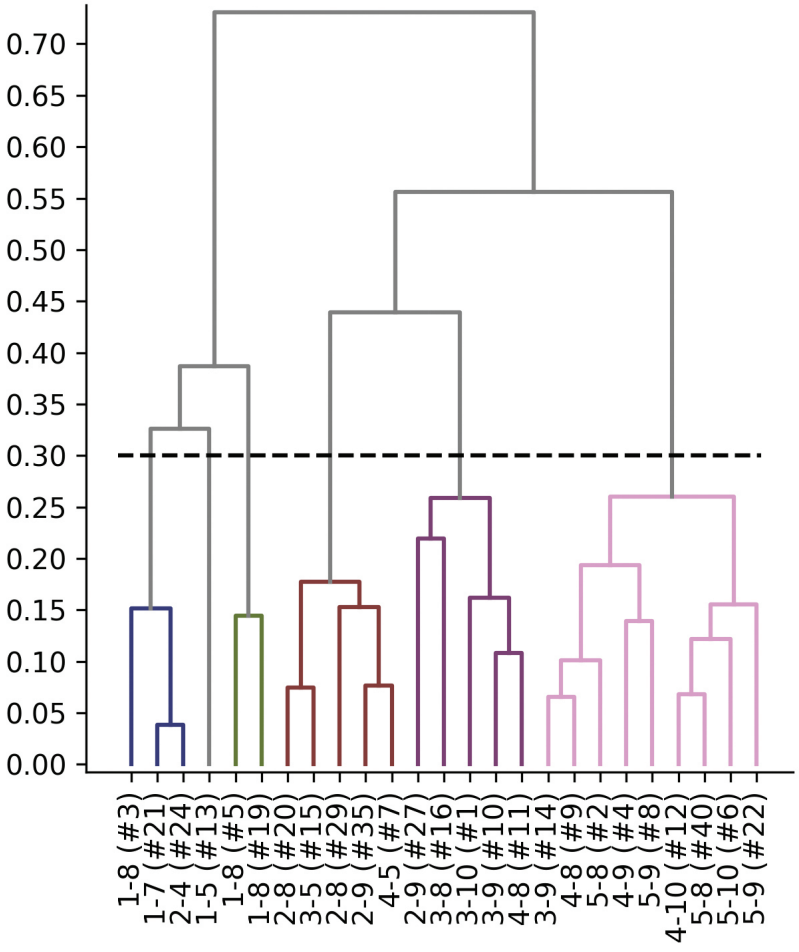


Fig. 3 The dendrogram of an agglomerative hierarchical clustering algorithm on the similarity matrix in Figure 2 with a threshold set of 0.7 similarity

The hierarchical clustering algorithm will always converge the network down to a single community. To be able to pull out N communities from the hierarchical clustering algorithm, one needs to use a dendrogram. A dendrogram is a way of pictorially viewing the merges that occurred during the hierarchical clustering algorithm and inspecting how the merges affect the similarity score of the community. Each horseshoe on the dendrogram represents a merger between the communities attached to the horseshoe. The height of the horseshoe depicts the increase in distance between the members of the community — the decrease in similarity. To determine the communities within the dendrogram, a line is draw at a given distance threshold. Each horizontal line that is cut through by the distance threshold is a community within the network. Figure 3 shows the dendrogram of using the above method on the similarity matrix from Figure 2. Based upon the work by Hughes et al. [16] an approximate threshold has been set of 0.7 similarity — 0.3 distance — to determine the communities within the network.

The six populations — from the associated communities — calculated from the dendrogram, break out of the prescribed populations based around a dominant feature of the structure and instead show previously unknown populations within the

network. Figure 4 shows the same heatmap as in Figure 2; however, in Figure 4 the structures are ordered in the x and y axis in community order. As one can see, once the ordering is changed to be by communities derived from the similarity metrics themselves, the similarity between the populations is clear.

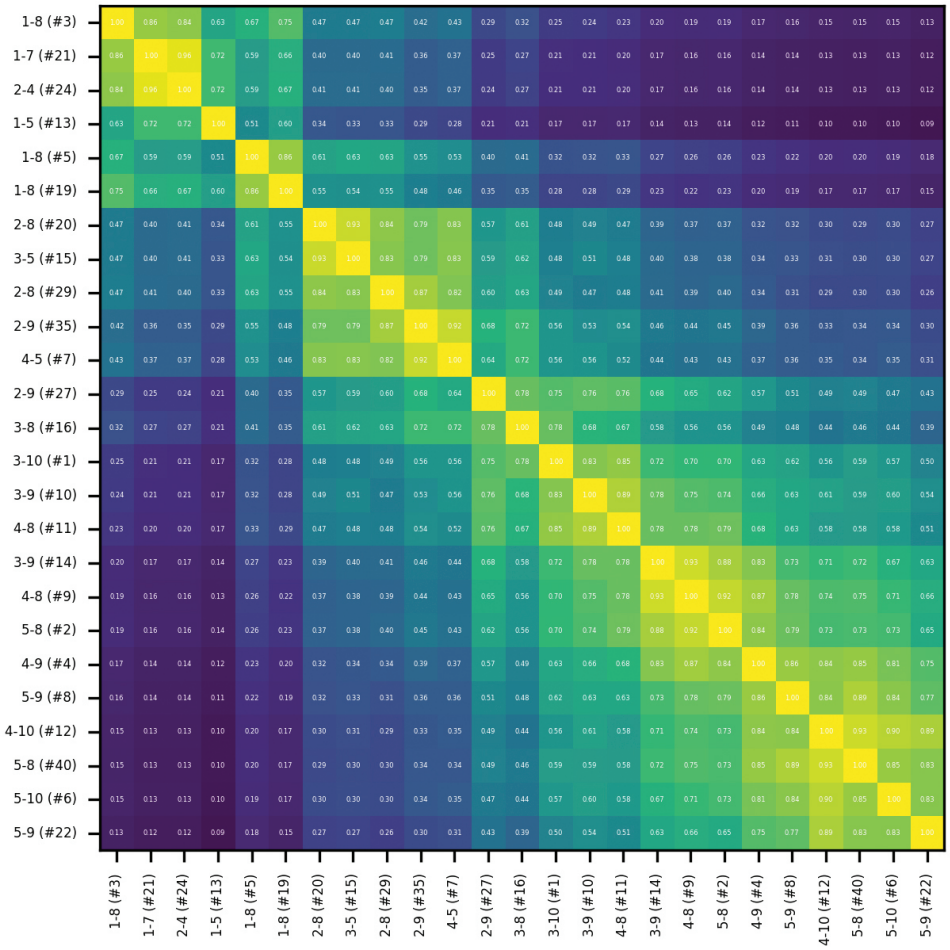


Fig. 4 NetPair similarity heatmap after reordering via the dendrogram using the first five bridges from each span in the beam-and-slab PEAR dataset

Conclusion

In conclusion, this paper has explored how to find network communities from a similarity metric generated by the PBSHM Framework. This paper has shown that our previous methods of using a distinct shared feature on the structures in question — to identify what a sub-population should be — is omitting hidden clusters of similar structures. Instead, letting the network of structures itself inform us on what communities exist, facilitates previously unknown clusters of similar structures being brought into focus. The hope is that by highlighting these unknown clusters of structures, one can potentially perform transfer learning on previously unexplored structures.

Acknowledgments The authors of this paper gratefully acknowledge the support of the UK Engineering and Physical Sciences Research Council (EPSRC) via grant reference EP/W005816/1. For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

References

1. L.A. Bull, P.A. Gardner, J. Gosliga, T.J. Rogers, N. Dervilis, E.J. Cross, E. Papatheou, A.E. Maguire, C. Campos, and K. Worden. "Foundations of Population-based SHM, Part I: Homogeneous populations and forms". *Mechanical Systems and Signal Processing*, 148:107141, 2021.
2. J. Gosliga, P.A. Gardner, L.A. Bull, N. Dervilis, and K. Worden. "Foundations of Population-based SHM, Part II: Heterogeneous populations – Graphs, networks, and communities". *Mechanical Systems and Signal Processing*, 148:107144, 2021.
3. P. Gardner, L.A. Bull, J. Gosliga, N. Dervilis, and K. Worden. "Foundations of Population-based SHM, Part III: Heterogeneous populations – Mapping and transfer". *Mechanical Systems and Signal Processing*, 149:107142, 2021.
4. George Tsialiamanis, C Mylonas, E Chatzi, Nikolaos Dervilis, David J. Wagg, and Keith Worden. "Foundations of Population-based SHM, Part IV: Structures and Feature Spaces as Geometry". *Mechanical Systems and Signal Processing*, 157:107692, 2021.
5. Daniel S. Brennan, Julian Gosliga, Elizabeth J. Cross, and Keith Worden. "Foundations of population-based SHM, Part V: Network, framework and database". *Mechanical Systems and Signal Processing*, 223:111602, 2025.
6. Paul Jaccard. "Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines". *Bull Soc Vaudoise Sci Nat*, 37:241–272, 1901.
7. Coen Bron and Joep Kerbosch. "Algorithm 457: Finding all cliques of an undirected graph". *Communications of the ACM*, 16(9):575–577, 1973.
8. Chandula T Wickramarachchi, Julian Gosliga, Andrew Bunce, Daniel S Brennan, David Hester, Elizabeth J Cross, and Keith Worden. "Similarity assessment of structures for population-based structural health monitoring via graph kernels". *Structural Health Monitoring*, page 14759217241265626, 2024.
9. Daniel S. Brennan, Timothy J. Rogers, Elizabeth J. Cross, and Keith Worden. "On calculating structural similarity metrics in population-based structural health monitoring". *Data-Centric Engineering*, 6:e24, 2025.
10. Daniel S. Brennan, Timothy J. Rogers, Keith Worden, and Brennan Klein. "Structural motifs as signatures for distinguishing critical infrastructure networks". *Submitted to Nature Communications*.
11. Simon Rogers and Mark Girolami. *A First Course in Machine Learning*. Chapman & Hall/CRC Machine Learning & Pattern Recognition Series. CRC Press, Taylor & Francis Group, Boca Raton London New York, second edition, first issued in paperback edition, 2020.
12. Albert-László Barabási and Márton Pósfai. *Network Science*. Cambridge University Press, Cambridge, United Kingdom, 2016.
13. Mark E. J. Newman. *Networks*. Oxford University Press, Oxford, second edition edition, 2018.
14. Connor O'Higgins, Daniel S. Brennan, Tristan Gowdridge, Alan J. Ferguson, David Hester, and Keith Worden. "Population-based SHM Engineered Asset Resource (PEAR) database for testing and validation of PBSHM methods". *Submitted to Structural Health Monitoring*.
15. E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A.-L. Barabási. "Hierarchical Organization of Modularity in Metabolic Networks". *Science*, 297(5586):1551–1555, 2002.
16. A. J. Hughes, J. Poole, N. Dervilis, P. Gardner, and K. Worden. "Quantifying the Value of Information Transfer in Population-Based SHM". In *Data Science in Engineering Vol. 10*, pages 19–31, New York, 2025. River Publishers.

River Rapids

Data Science in Engineering, Volume 9

Thomas Matarazzo ■ François Hemez ■ Eleonora Maria Tronci ■ Austin Downey

Data Science in Engineering, Volume 9: Proceedings of the 44th IMAC, A Conference and Exposition on Structural Dynamics, 2026, the ninth volume of ten from the Conference, brings together contributions to this important area of research and engineering. The collection presents early findings and case studies on fundamental and applied aspects of Data Science in Engineering, including papers on:

- Novel Data-Driven Analysis Methods
- AI-Driven Digital Twins for Structural Modeling and Dynamic Characterization
- Transfer Learning and Population-Based Monitoring
- Data-Driven Techniques for System Prognostics and Health Monitoring
- Applications of AI in Structural Dynamics and System Identification
- Uncertainty Quantification in Data-Driven and Hybrid Models
- Physics Informed Machine Learning for Dynamic Systems

ISBN 978-87-438-1448-1



River Publishers