Physics-Informed Neural Network for Analyzing Elastic Beam Behavior

SOHEIL HEIDARIAN RADBAKHSH, KAMYAB ZANDI
and MAZDAK NIK-BAKHT

ABSTRACT

This paper introduces a methodology that combines a physics-based model with observed data for accurately modeling the deflection of an elastic beam in the context of structural health monitoring. The challenges associated with physics-based and data-based methods such as computational time, simplifying assumptions, and seamless integration of sensor data with physics-based models are addressed. The presented method offers a promising approach by effectively fusing data with prior physical knowledge in a cost-effective manner. The proposed methodology is validated through comparisons with analytical and finite element analysis methods for beams with various irregularities such as point loads and supports. The results demonstrate the advantages of integrating sensor data into the model for faster convergence and improved accuracy.

INTRODUCTION

The core of structural health monitoring (SHM) research lies in utilizing data collected from physical structures. This data provides valuable insights into the behavior of structures when subjected to different stimuli, facilitating continuous monitoring, prediction, and effective control and performance of structures during their operational lifetime [1]. In order to model and analyze complex structural system within the framework of SHM, either a data-based or physics-based model can be employed [2].

In recent years, data-based models have gained significant attention and are increasingly recognized as promising tools, particularly for modeling complex systems that involve model uncertainties or are usually presented with simplified relationships. Nevertheless, the effectiveness of data-driven modeling techniques heavily relies on the quantity and quality of the available data. They typically require a substantial amount of data to converge and provide reliable predictions. Another challenge arises from the lack of generalizability, as data-driven models may struggle to perform effectively when faced with new or unseen conditions. It should be noted that data-based models, despite their capabilities, inherently disregard physical laws which causes these issues [3].

Soheil Heidarian Radbakhsh and Mazdak Nik-Bakht, Dept. of Building, Civil, and Environmental Engineering, Concordia University, Montréal, QC H3G 1M8, Canada.
Kamyab Zandi, Dept. of Architecture and Civil Engineering, Division of Concrete Structures, Chalmers University of Technology, SE 412 96, Gothenburg, Sweden.

Physics-based models provide the advantage of being less dependent on the availability of extensive data. They can leverage fundamental physical principles to describe the behavior of complex systems. However, these models are often constrained by computational complexity and the challenge of accurately representing the full range of physics involved in the system. Computational demands and the need for comprehensive understanding of the underlying physics can limit the practicality and applicability of physics-based models in complex systems. Furthermore, integrating data from instrumented structures into physics-based models is challenging. This process entails estimating model parameters using measured response data and refining these parameters either directly or through iterative methods. Accurate results relies on the physics-based model closely resembling the actual response without excessive simplifications and on the selection of appropriate parameters for modeling [2].

The mentioned approaches have certain limitations that make them less suitable for complex structures in uncontrolled environments. This study presents a computationally affordable modelling approach for obtaining a representative of a system response that balances the information from the observed data and the physics-based model. The model utilizes the observed data to improve the physics-based model while guiding the model in unmeasured areas of the structural domain.

LITERATURE REVIEW

In recent years, there has been significant interest in physics-informed machine learning (PIML) [4] and their different variations [1], [5]. These approaches have introduced a new paradigm and perspective on fusing machine learning with physics knowledge, offering potential solutions to overcome the shortcomings of the mentioned methods.

The integration of physics with machine learning, aiming to bridge the gap between data-driven and physics-based approaches, develops diverse methodologies, referred to as physics-informed or -guided machine learning methods. When referring to physics it means incorporating a mathematical equation that represents the underlying physics of the problem. However, these approaches differ in their approaches to integrate observed data with the governing physics. One approach involves using physics models to simulate data by executing physics-based models across various input combinations for training a machine learning model and leverage the embedded knowledge in the physics models [6], [7] (Figure 1.a). The second approach, showed in Figure 1.b, resembles discrepancy modeling, where a discrepancy term is introduced to compensate for the differences between the incomplete/simplified physics and the actual system behavior [8]. In this study, the physics-informed neural networks (PINNs) refer to a method that integrates the physics law as a penalization term into the loss function, introduced by Raissi et al. (2019) [5]. This approach, Figure 1.c represents another means of employing PINNs, distinct from the previously discussed methods involving data simulation and discrepancy modeling.

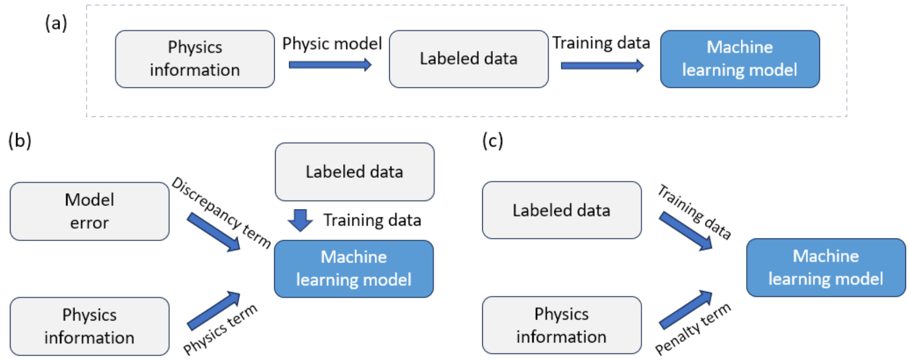

Figure 1. Different PINN options to combine physics information and observation data.

PINN was investigated by many researchers in the context of structural engineering applications, showing promising potential in various areas, including (SHM) [8]. In the domain of structural modeling and analysis, the physics-based aspect of the PINN incorporates beam theory equations [9] and the equation of motion [8] for single and multi-degree freedom systems (SDOF and MDOF). The machine learning component employs various networks, such as Artificial Neural Network (ANN) [5], long short term memory (LSTM) [10] and convolutional neural network (CNN) [11].

In the context of SHM, Yuan et al. (2020) proposed a new framework by integrating physics knowledge into loss function of neural network to simulate structural systems based on Euler-Bernoulli [3]. There are several studies trying implementing PINNs [9], [12] with the beam theory in the context of structural system modeling. However, none of them considered irregularities such as point load and supports in beam length. The goal of the study is to model the transverse displacement of the beam from integration of physics knowledge and the boundary conditions and sensory data to make a seamless integration between observed data and the model considering irregularities.

PINN ARCHITECTURE AND IMPLEMENTATION

This section provides an overview of the proposed methodology for modeling an elastic beam. Initially, the PINN architecture and the training process specific to beam modeling are described. Subsequently, the methodology employed to address irregularities, such as point loads or supports, along the beams' length is explained. PINNs are trained by minimizing a combination of loss functions. To train the model, the mean square error loss for a neural network is minimized, comparing the predicted and actual solutions. Furthermore, additional loss terms are introduced to account for deviations from established physical laws. The integration of the physical part involves formulating the physics equation derived from the neural network. By utilizing automatic differentiation (AD), the derivatives of the network outputs, such as deflection, with respect to the network inputs, are computed and enables the formation of the equation.

In this research, DeepXDE [13] with a Tensorflow [14] backend was utilized. DeepXDE is an accessible PINN solver which handles various types of differential equations, including ordinary differential equations (ODEs) and partial differential equations (PDEs), and supports complex domain geometries using constructive solid geometry. Further details on DeepXDE and its utilization are explained in [13].

Training procedure of PINN for beams

Euler-Bernoulli beam equation was chosen as the underlying physics equation in this study. The equation has been well-established in previous research [9], [12] and serves as an appropriate example of incomplete prior knowledge. For the machine learning component, a feedforward neural network (NN) with multiple hidden layers is employed to approximate the equation's solution. Previous investigations have demonstrated that feedforward NNs are effective in solving a wide range of differential equations [13]. Euler-Bernoulli equation (Equation 1) models the deflection characteristics of beams, y, in the space domain subject to the transverse loading.

$$\frac{d^2\left(EI(x)\left(\frac{d^2y(x)}{dx^2}\right)\right)}{dx^2} = q(x). \tag{1}$$

The parameter *EI* is flexural rigidity, $\frac{d^4y}{dx^4}$ represents the fourth order partial derivative of y with respect to x, and q is distributed loading. This work considers a uniform cross-sectioned beam with constant material properties throughout the beam.

In Figure 2, the utilized PINN architecture for solving the Euler-Bernoulli beam equation is illustrated. On the left side, the neural network as a function approximator for the solution of the problem is depicted. The right-hand side corresponds to the residual of the differential equation, obtained by applying the AD to the neural network and encoding the partial differential equation into the algorithm's architecture. At the end the loss function is formed by the weighted summation of two parts, having the same parameters (weights and biases). There is an extra term added to the conventional PINN architecture which will be explained in the next section. The middle section of the figure depicts AD which is derived by applying the chain rule to calculate the derivatives of the output with respect to the input to determine the losses of equation terms, as well as the terms of the high order derivative boundary conditions, such as slope and moment.

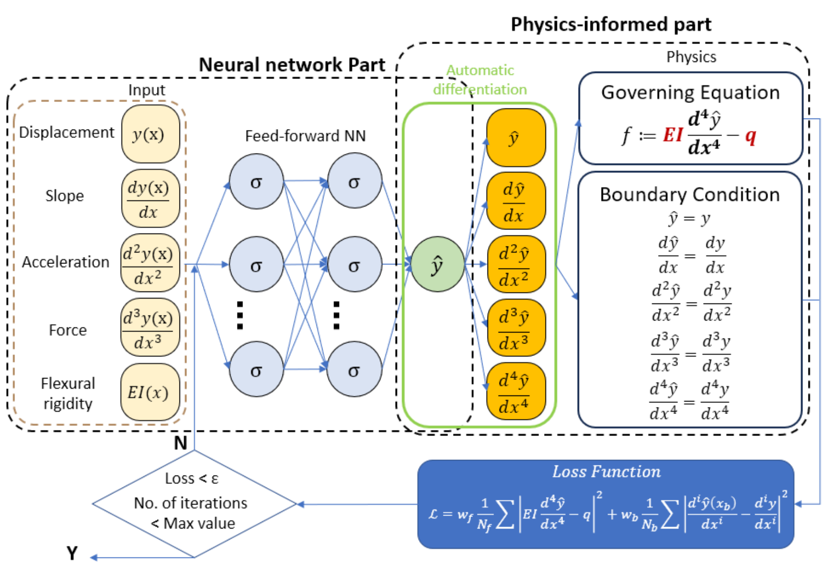

Figure 2. Schematic of PINN framework for solution of Euler-Bernoulli beam.

$$\mathcal{L} = w_f\mathcal{L}_f + w_b\mathcal{L}_b; \quad \mathcal{L}_f = \frac{1}{N_f}\sum\left|EI\frac{d^4\hat{y}}{dx^4} - q\right|^2; \quad \mathcal{L}_b = \frac{1}{N_b}\sum\left|\frac{d^i\hat{y}(x_b)}{dx^i} - \frac{d^iy}{dx^i}\right|^2 \tag{2}$$

By minimizing the loss function of the defined algorithm, the model will be trained, which is written in Equation 2. Where w_f and w_b represent the weights assigned to the differential equation and boundary condition losses, respectively. While it is feasible to use different weights for these components, it is assumed that these weights are equal for the purpose of this study.

Training PINN for beams with irregularity

When dealing with irregularities along the beam certain considerations come into play. Initially, the solution for concentrated loads is explained, and then it is expanded to encompass cases involving supports along the beam's length. When the concentrated load is applied at an arbitrary location along the beam's span, rather than at the ends, the right-hand side of the Euler-Bernoulli beam equation (Equation 1) becomes zero due to the concentrated nature of the load. While the deflection is continuous along the beam, a discontinuity is created in the third derivative of the deflection, responding to the force.

To address this scenario, a novel solution inspired by a numerical approach for handling beam equations with concentrated loads is proposed. This approach is anticipated to yield more generalized solutions. The beam is divided into two segments: one to the left of the applied load and the other to the right. For each segment, the governing differential equation of the problem is defined using the Euler-Bernoulli beam equation with zero distributed loads. Then, a new set of boundary conditions is established at the point where the load is applied which are named interface conditions. This methodology allows us to address irregularities effectively, providing a versatile solution to various scenarios involving concentrated loads or supports.

In the case of a point load, the deflection, y, and its first and second derivatives exhibit continuity at the point load location. However, the force, i.e., the third derivative of deflection, experiences a discontinuity equivalent to the magnitude of the point load. To account for these conditions, they are incorporated as interface conditions in the loss function. Similarly, in the case of supports, while the magnitude of the force is unknown, the value of the deflection is known and can be utilized as an additional condition. Thus, when irregularities occur, the domain is decomposed. However, decomposition resulting in certain costs, which will be elaborated on in the following section.

RESULTS AND DISCUSSION

In this section, the linear deflection of an elastic beam with irregularity and different boundary conditions is investigated while fusing the sensor data to the model. The properties of beams are $E = 24.9\text{GPa}$, $I = 4.5 * 10^4\text{m}^4$ with the beam length being 5m. Moreover, in all cases considered, the loadings are $5 * 10^5\text{N}$ and $5 * 10^5\text{N/m}$ for point and distributed loads, respectively.

TABLE I. PINN HYPER-PARAMETERS.

Parameters	No. of training points	Hidden layers	No. of node	Learning rates	Optimizer	Activation function
Value	20	3	20	0.0005	Adam	Tanh

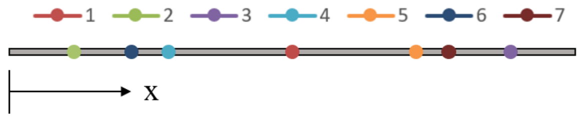

Figure 3. Schematic of the elastic beam with the location of observed data.

The neural network architecture and hyperparameters were selected through a systematic trial-and-error approach, prioritizing both solution accuracy and computational efficiency for all hyperparameters except for the number of iterations, as a computational cost indicator for measuring the effect of auxiliary data. All used hyperparameters are indicated in Table I. For the number of iterations, the examination was conducted up to 150,000 iterations, but beyond 40,000 iterations, the observed improvement in solution accuracy was not significant. Similar initial weights and biases across all models facilitate a fair and valid comparison when incorporating sensor data.

The results of models were validated with analytical and finite element analysis methods (through Wolfram Mathematica; and SAP2000 and Ansys respectively). Figure 3 illustrates the location of data points that are used as auxiliary data representing response data collected from sensors on the beam with the coordinates along the x axis being [2.5, 0.625, 4.375, 1.25, 3.75, 1.875, 3.125] from 1 to 7 (units are in meter). For simulating sensors data, the data extracted from Ansys model is used.

To assess the accuracy of the trained model, the relative percentage error is employed as an error estimation measure. However, instead of considering the error at a single midpoint like Kapoor et al. (2023), the cumulative error of 10 points distributed along the beam span is utilized. In eq.3 , u^* is the prediction and u is the actual solution [12].

$$Error = \frac{u^* - u}{u} * 100 \tag{3}$$

The study investigated two types of irregularities in the beam. The first being the presence of point loads or supports along its length, which can complicate beam modeling. Beams with different point loads and support conditions are tested. Additionally, beams with both clamped and hinged supports were modeled to showcase the capability of PINN under different boundary conditions. Initially, all the models are trained solely using boundary conditions, and subsequently, actual data is incorporated into the models. Figure 4 depicts the comparison between the error (deviation from actual behavior) and the number of iterations (shown in logarithmic scale), which serves as an indicator of computational cost.

The results of the comparison demonstrate that incorporating sensor data into the model aids in faster convergence towards the solution. For example, when considering three-point loads and running 1000 iterations, the addition of a single data point reduces the error by 75%. However, when examining the effect of adding sensor data on reducing error with fewer iterations, no clear trend emerges, necessitating further investigations. Additionally, it is observed that irregularities with supports result in higher errors, potentially attributable to the more complex shape associated with such irregularities.

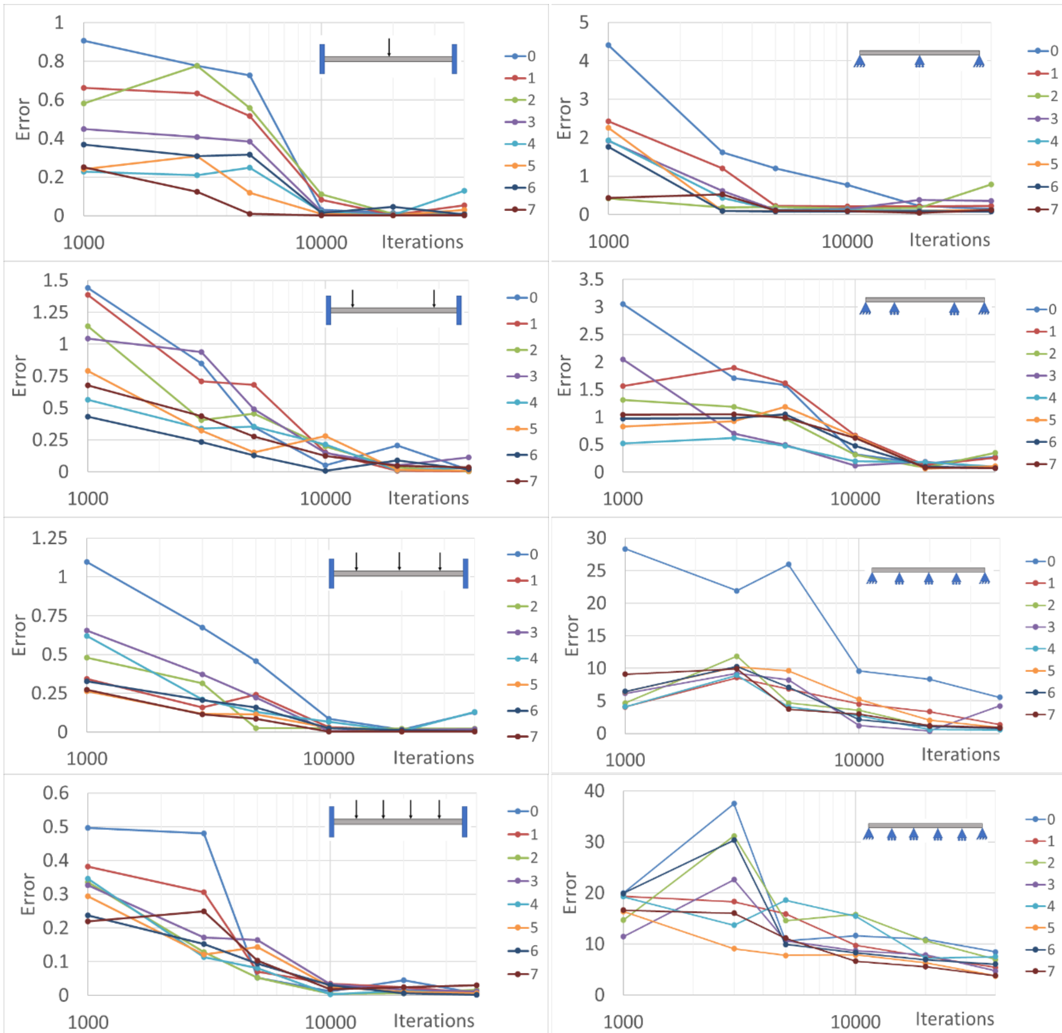

Figure 4. Comparison of beam accuracy with added sensor data to the trained model.

CONCLUSION

This paper has presented a physics-informed neural network model that fuses together information from a physics-based model and observed data for modeling the deflection of an elastic beam. The observed data collected from sensor networks, as the response of beam, can be in the form of stress, vibration/acceleration/moment, slope, and deflection. The presented method eases the utilization of measured data from structure into physics-based model.

The main challenges in structural analysis and modeling in the context of SHM encompass computational time and incorporating too many simplifying assumptions. Moreover, integrating sensor data with physics-based models poses limitations due to the lack of seamless integration. On the other hand, acquiring enough data is challenging and expensive. However, Physics-Informed Neural Networks (PINNs) offer a simulation-based, computationally efficient, and cost-effective alternative by effectively integrating incomplete or noisy information with existing physical knowledge.

REFERENCES

1. A. Gregory, F. D.-H. Lau, M. Girolami, L. J. Butler, and M. Z. E. B. Elshafie, “The synthesis of data from instrumented structures and physics-based models via Gaussian processes,” *Journal of Computational Physics*, vol. 392, pp. 248–265, Sep. 2019, doi: 10.1016/j.jcp.2019.04.065.
2. S. Heidarian Radbakhsh and Nik-Bakht, Mazdak, “Model Updating for Structural Digital Twins through Physics-Informed Data-Driven Models,” presented at the CSCE, Moncton, 2023.
3. F.-G. Yuan, S. Zargar, Q. Chen, and S. Wang, “Machine learning for structural health monitoring: challenges and opportunities,” Apr. 2020, p. 2. doi: 10.1117/12.2561610.
4. G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, “Physics-informed machine learning,” *Nat Rev Phys*, vol. 3, no. 6, pp. 422–440, Jun. 2021, doi: 10.1038/s42254-021-00314-5.
5. 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*, vol. 378, pp. 686–707, Feb. 2019, doi: 10.1016/j.jcp.2018.10.045.
6. K. Fukui, J. Tanaka, T. Tomita, and M. Numao, “Physics-Guided Neural Network with Model Discrepancy Based on Upper Troposphere Wind Prediction,” in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, Dec. 2019, pp. 414–419. doi: 10.1109/ICMLA.2019.00078.
7. T. G. Ritto and F. A. Rochinha, “Digital twin, physics-based model, and machine learning applied to damage detection in structures,” *Mechanical Systems and Signal Processing*, vol. 155, p. 107614, Jun. 2021, doi: 10.1016/j.ymssp.2021.107614.
8. Z. Lai, C. Mylonas, S. Nagarajaiah, and E. Chatzi, “Structural identification with physics-informed neural ordinary differential equations,” *Journal of Sound and Vibration*, vol. 508, p. 116196, Sep. 2021, doi: 10.1016/j.jsv.2021.116196.
9. M. Bazmara, M. Silani, M. Mianroodi, and M. sheibanian, “Physics-informed neural networks for nonlinear bending of 3D functionally graded beam,” *Structures*, vol. 49, pp. 152–162, Mar. 2023, doi: 10.1016/j.istruc.2023.01.115.
10. R. Zhang, Y. Liu, and H. Sun, “Physics-informed multi-LSTM networks for metamodeling of nonlinear structures,” *Computer Methods in Applied Mechanics and Engineering*, vol. 369, p. 113226, Sep. 2020, doi: 10.1016/j.cma.2020.113226.
11. Y. Bao and S. Mahadevan, “Diagnosis of interior damage with a convolutional neural network using simulation and measurement data,” *Structural Health Monitoring*, vol. 21, no. 5, pp. 2312–2328, Sep. 2022, doi: 10.1177/14759217211056574.
12. T. Kapoor, H. Wang, A. Nunez, and R. Dollevoet, “Physics-informed neural networks for solving forward and inverse problems in complex beam systems.” arXiv, Mar. 02, 2023. doi: 10.48550/arXiv.2303.01055.
13. L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, “DeepXDE: A Deep Learning Library for Solving Differential Equations,” *SIAM Rev.*, vol. 63, no. 1, pp. 208–228, Jan. 2021, doi: 10.1137/19M1274067.
14. M. Abadi *et al.*, “TensorFlow: a system for large-scale machine learning,” in *Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation*, in OSDI’16. USA: USENIX Association, Nov. 2016, pp. 265–283.