13th International Conference on
Structural Health Monitoring of Intelligent Infrastructure
DOI: 10.3217/978-3-99161-057-1-045

Developing physics-informed neural networks for structural parameters identification of beam with moving loads

Anmar I. F. Al-Adly[1], ORCID , Prakash Kripakaran[1], ORCID
[1]Department of Engineering, Faculty of Environment, Science and Economy, University of Exeter, Stocker Road, Exeter, EX4 4HF, United Kingdom, England
email: aa1224@exeter.ac.uk, P.Kripakaran@exeter.ac.uk

ABSTRACT: Physics-Informed Neural Networks (PINNs) seamlessly integrate the predictive capabilities of neural networks with established physical principles. By integrating constraints such as displacement and force boundary conditions alongside governing equations, PINNs can generate digital twins of physical systems and processes. This fusion allows for more accurate modelling and simulation of complex physical phenomena, bridging the gap between data-driven approaches and traditional physics-based methods. Nevertheless, the practical implementation of PINNs remains challenging, primarily due to numerous influential hyperparameters and the complex nature of modelling the governing physics through partial differential equations (PDEs). This challenge becomes especially critical in the context of dynamic loads, where higher-order PDEs encompassing both spatial and temporal domains, alongside relevant structural parameters and generalised (distributed) load's function, must be carefully optimised during the PINNs training process. This study presents a novel application of PINNs model, developed, trained, and validated using real-world bridge monitoring data, for the inverse problem of predicting structural parameters of a girder subjected to moving loads. Two case studies are considered. In the first, PINNs model is utilised to estimate the structural parameters of a bridge girder under varying levels of noise in the data. In the second, the model is trained with actual field monitoring measurements to estimate structural parameters while predicting girder deflection and other internal forces. The findings advance the existing body of knowledge in structural health monitoring (SHM) by demonstrating a practical PINNs-based solution for bridge girders under moving loads.

KEY WORDS: PINNs; Inverse problem; Parameter identification; Neural networks (NNs); SHM.

1 INTRODUCTION

Bridges are critical components of transportation infrastructure, and ensuring their safety and longevity through structural health monitoring (SHM) is a top priority [1]. Over time, bridges are subjected to environmental deterioration and repetitive traffic loads that can induce damage or stiffness degradation, potentially reducing their service life [2]. A key aspect of SHM is structural parameter identification – determining properties such as stiffness (flexural rigidity) – which enables the assessment of a structure's condition [3].

Recently, physics-informed neural networks (PINNs) have emerged as a promising tool for tackling inverse modelling problems [4], [5]. PINNs are a type of deep learning framework that embeds physical laws—often expressed as partial differential equations (PDEs)—into the neural network's training process. Instead of relying solely on labelled input-output pairs, PINNs are trained to predict outputs that both fit observed data and satisfy governing physics. This is achieved by incorporating physics-based constraints, such as PDE residuals, into the loss function alongside data-fitting terms [6]. For instance, PINNs have been employed to enforce structural dynamics and damping evolution equations while matching measured responses [3].

In the context of inverse problems, the primary objective of PINNs is to estimate unknown parameters within partial PDEs based on observed data. PINNs have been applied to various domains, including fluid dynamics, where they determine density, velocity, and pressure fields for one-dimensional Euler equations from observed density gradient data [7]. Rasht et al.[8] utilised PINNs for seismic imaging, estimating wave speed from observed data in full waveform inversions. Current advancements have further extended the application of PINNs to SHM, particularly in civil engineering structures such as bridges and beams. For example, in the context of railway bridges, PINNs have been employed for structural analysis and monitoring, addressing challenges such as load distribution modelling, SHM, and failure prediction under dynamic train loads [9]. Moreover, studies have demonstrated the ability of PINNs to predict structural response from sparse sensor data, even in the presence of noise for Kirchhoff–Love plates under static loads [10]. These applications highlight the capability of PINNs to integrate data fidelity with physical consistency, making them particularly suitable for SHM tasks, where field measurements are often limited.

However, a key challenge lies in applying PINNs to identify structural parameters of beams subjected to moving loads using real-world data. This research aims to address this challenge by proposing a novel approach that integrates physics-based modelling with real-world monitoring data to improve the accuracy and reliability of structural parameter identification.

The main contributions of this study are as follows:

- The development, training, and validation of PINNs to solve the problem of a bridge girder subjected to a moving load. In doing so, this work addresses the challenge of incorporating higher-order PDEs spanning both spatial and temporal domains.

DOI: 10.3217/978-3-99161-057-1-045

- Demonstration of how PINNs can predict structural parameters, alongside other outputs such as deflection and its derivatives.
- A thorough examination of the PINNs' performance and predictive accuracy under varying levels of data noise. By testing the models across a range of signal-to-noise ratios, the study evaluates their robustness, providing insights into their reliability and resilience in practical settings where data imperfections are common.
- An investigation of the PINNs' adaptability and generalisation capabilities when trained on real-world monitored data from a bridge.

This paper is organised as follows: Section 2 introduces the problem definition and outlines the methodology. Section 3 provides discussions on Case Studies 1 and 2, along with their respective results. Section 4 presents the conclusions and recommendations for future research.

2 METHODOLOGY

This section considers the architecture of a fully connected feedforward neural network and describes the PINNs setup for inverse problems. Specifically, it includes an explanation of PINNs model hyperparameters and the loss function terms for a bridge girder with a moving load. The analysis includes a detailed discussion on the training of PINNs, as well as the investigated and selected hyperparameters of the neural networks.

2.1 Neural networks Architecture

The network consists of hidden layers (H), each formulated to process inputs recursively. For each hidden layer *h*, where h ranges from 1 to H, the output *f(h)* is determined by Equation (1):

$$f(h) = \sigma\big(W(h)f(h-1) + b(h)\big) \tag{1}$$

Here, $f(0)$ signifies the initial input to the neural network. The function $\sigma(\cdot)$ denotes the nonlinear activation function, which is crucial for enabling the network to model complex patterns. The weight matrix $W(h)$ and the bias vector $b(h)$ pertain to the h −th hidden layer respectively [11].

The output of the neural network, denoted as $u\ (i.e., beam's\ deflection)$, is derived from the last hidden layer's output through the Equation (2):

$$u = W(H+1)f(H) + b(H+1) \tag{2}$$

In this formulation, $W(H+1)$ and $b(H+1)$ represent the weight matrix and bias vector of the output layer, respectively. The entire set of parameters within the network, which includes all the weights and biases for each layer, is collectively denoted by (θ) as listed in Equation (3):

$$\theta = \{W(1), b(1), \dots, W(H), b(H), W(H+1), b(H+1)\} \tag{3}$$

This parameter set θ encapsulates all elements necessary for the neural network's function, facilitating a unified approach to training and adjustment during the learning process.

The structure of this neural network combines linear operations and nonlinear activation functions. Utilising activation functions that are infinitely differentiable, such as hyperbolic functions, the architecture enables the calculation of derivatives of any order for the output relative to the input of the neural network via automatic differentiation [4], [5]. These derivatives are instrumental in incorporating basic physical principles into the loss functions used in PINNs.

2.2 Framework of PINNs for inverse problems

PINNs presents an approach for addressing forward and inverse problems in partial differential equations (PDEs) across various systems [4]. A typical inverse problem for a simple beam (girder) subjected to a moving can be described by the following formulation:

1. Governing Equation: The Bernoulli-Euler beam theory [12] is widely used to describe beam behaviour under bending. This theory neglects shear deformation and rotational effects [13], which are typically minor in mostly bending conditions. Previous studies [14], [15] have numerically modelled bridges using this theory, focusing on moving loads that simulate vehicular traffic. Therefore, this paper briefly presents only the essential PDE needed for developing PINNs.

For a linearly elastic beam, the vertical deflection, $u(x,t)$ along the *z*-direction satisfies the PDE shown in Equation (4) [14], [16]:

$$EI\frac{\partial^4 u(x,t)}{\partial x^4} + \mu\frac{\partial^2 u(x,t)}{\partial t^2} = P\delta(x-vt) \tag{4}$$

Where: $\frac{\partial^4 u(x,t)}{\partial x^4}$ is the beam's curvature under bending, *EI* is the flexural rigidity, $\frac{\partial^2 u(x,t)}{\partial t^2}$ represents beam acceleration due to vibration, μ is the constant mass per unit length, *P* is the moving load magnitude, v is the load's constant speed, δ is the Dirac delta function modelling the instantaneous position of the moving load.

This paper employs a Gaussian-based approach to approximate the Dirac delta function, leveraging its smoothness and regularity [17]. The approximation is defined by Equation (5):

$$P\delta(x-vt) \approx \frac{1}{\beta\sqrt{\pi}} e^{-\frac{(x-vt)^2}{\beta^2}} as \beta \to 0 \tag{5}$$

Where β the regularisation parameter governs the approximation's smoothness and accuracy. The vt term specifies the vehicle's position as the product of speed and time since it entered the structure.

2. Initial (IC) and Boundary (BC) conditions: The beam is assumed to be at rest initially. This is given in mathematical form in Equations (6) and (7). Equation (6) indicates that vertical deflection $u(x,t)$ is zero along the whole length of the beam at T = 0, while Equation (7) indicates the beam is perfectly still, experiencing no vibration, at T = 0 [14], [16].

$$u(x,t) = 0 \ \ for\ x \in [0,L], \qquad T = 0 \tag{6}$$

$$\frac{\partial u}{\partial t}(x,t) = 0 \ \ for\ \ x \in [0,L], \qquad T = 0 \tag{7}$$

DOI: 10.3217/978-3-99161-057-1-045

For a simple beam supports at its two ends. The beam cannot move vertically but is free to rotate about y-axis at these locations, namely at $x = 0$ and $x = L$. The latter necessitates that the corresponding force quantity - the bending moment, which is directly proportional to the beam curvature and computed as the second derivative of $u(x,t)$, must always be zero at $x = 0$ and $x = L$. Equations (8), and (9) mathematically represent these boundary conditions.

$$u(0,t) = 0, and\ u(L,t) = 0,\ \ for\ \ \in [0,T] \tag{8}$$

$$\frac{\partial^2 u}{\partial x^2}(0,t) = 0, and\ \frac{\partial^2 u}{\partial x^2}(L,t) = 0, for\ t \in [0,T] \tag{9}$$

3. Loss function: In this study, we develop PINNs to predict structural parameters and internal forces—such as deflection and moment—in a bridge girder under a moving load of known magnitude and constant speed. Unlike traditional neural networks, which minimise the mean squared error (MSE) using training data, PINNs incorporate additional loss terms to enforce the governing partial differential equation (PDE), initial conditions, and boundary conditions, ensuring physically consistent predictions [18]. Accordingly, let us assume that γ represents parameters related to the physics of the system—some or all of which may be unknown and must be estimated in inverse problems. To achieve this, PINNs are trained to minimise a composite loss function ($\mathcal{L}$), which can be expressed as in Equation (10).

$$L(\Theta) = \alpha_f L_f(\Theta) + \alpha_i L_i(\theta) + \alpha_b L_b(\theta) + \alpha_d L_d(\theta) \tag{10}$$

Where $\Theta = \{\theta, \gamma\}$, with θ as neural network parameters defined in Equation (3) and γ as unknown physical parameters to be estimated (EI, flexural rigidity), $\mathcal{L}_f$ for the PDE, $\mathcal{L}_i$ for the initial conditions, $\mathcal{L}_b$ for the boundary conditions, and $\mathcal{L}_d$ for the measured data. Each term is scaled by a corresponding weight—α_f, α_i, α_b, and α_d. With prediction of PINNs, $u(x,t;\theta)$, the individual loss terms are expressed by Equations (11-15):

$$\mathcal{L}_f(\Theta) = \frac{1}{N_f}\sum_{i=1}^{N_f} \|F[g(x_i,t_i;\theta);\gamma]\|^2 \tag{11}$$

$$g(x_i,t_i;\theta) = EI\frac{\partial^4 u(x_i,t_i;\theta)}{\partial x^4} + \mu\frac{\partial^2 u(x_i,t_i;\theta)}{\partial t^2} - P\delta(x_i - vt_i) \tag{12}$$

$$\mathcal{L}_i(\theta) = \frac{1}{N_i}\sum_{i=1}^{N_i}\left\|u(x_i,t_i;\theta) + \frac{\partial u(x_i,t_i;\theta)}{\partial t}\right\|^2 \tag{13}$$

$$\mathcal{L}_b(\theta) = \frac{1}{N_b}\sum_{i=1}^{N_b}\left\|u(x_i,t_i;\theta) + \frac{\partial^2 u(x_i,t_i;\theta)}{\partial x^2}\right\|^2 \tag{14}$$

$$\mathcal{L}_d(\theta) = \frac{1}{N_d}\sum_{i=1}^{N_d}\|D(x_i,t_i;\theta) - u(x_i,t_i;\theta)\|^2 \tag{15}$$

Where $N_f, N_i, N_b,$ and N_d are the number of data points (collocation points) in training datasets:

- PDE residual: $\{(x_i,t_i):\ 0 \le i \le N_f,\ \ 0 \le x_i \le L,\ 0 \le t_i \le T\}$
- Initial condition: $\{(x_i,\ \ 0):\ 0 \le i \le N_i,\ \ 0 \le x_i \le L\}$,
- Boundary condition:$\{(0,t_i) and (L,t_i): 0 \le i \le 2N_b, 0 \le t_i \le T\}$, and
- Data points: $\{(x_i,t_i):\ 0 \le i \le N_d,\ \ 0 \le x_i \le L, 0 \le t_i \le T\}$, respectively.

The minimiser of $\mathcal{L}(\Theta)$ can be expressed by Equation (16):

$$\Theta^* = arg\ \min_{\Theta} \mathcal{L}(\Theta) \tag{16}$$

where $\Theta^* = \{\theta^*, \gamma^*\}$ with θ^* and γ^* being the optimal parameters and the estimated unknown parameters, respectively, and *arg min (·)* is the arguments of the minimum.

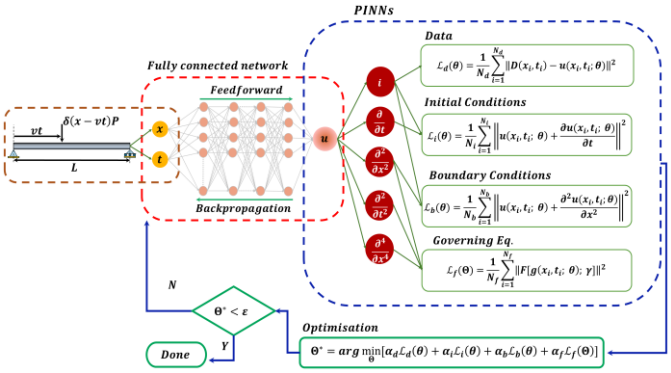


Figure 1.Physics-informed neural networks (PINNs) Schematic for an inverse problem of a beam with moving loads.

2.3 Training of PINNs

Figure 1 presents a schematic overview of the PINNs framework configuration. The setup consists of a fully connected feed-forward neural network (NN) that takes spatial and temporal inputs *(x, t)* to predict the solution *u (x, t)*—representing the expected deflection at a given location and time on the bridge's girder—and to estimate unknown physical structural parameters (γ). The training process is guided by the composite loss function defined in Equation (10) and optimised as shown in Equation (16). The training of the PINN is guided by the combined loss function given in Equation (10) and optimised as showed in Equation (16). Training the PINNs involves careful tuning of various hyperparameters, including the number of hidden layers, the number of neurons per layer, the choice of activation functions, the number of collocation points in the training dataset, optimiser configurations, and loss weighting factors.

Table 1 provides a summary of these hyperparameters, detailing their respective ranges and final selected values. The NN architecture, comprising five hidden layers and 64 neurons per layer, was chosen after testing various configurations. The Tanh activation function was selected due to its non-linear characteristics, outperforming others such as ReLU, SiLU, and Sigmoid. The collocation points, set at 2500 for the PDE, 2000

DOI: 10.3217/978-3-99161-057-1-045

for the boundary conditions, and 1000 for the initial conditions, provide a balance between computational efficiency and the resolution needed to accurately enforce the underlying physics. Adam optimiser was employed for its reliable convergence characteristics, and the learning rate was scheduled to decrease quadratically with epochs, starting from an initial value of 10^{-2}. Additionally, an adaptive loss-weighting strategy was used to balance the various terms, scaling each component relative to the smallest observed loss. Furthermore, the number of training iterations (epochs) was chosen to be 60,000, from a tested range of 10,000–100,000, to achieve robust convergence and accurate results.

The code presented in Table 2 is written in Python [19] and leverages PyTorch [20], a robust deep learning library. The PINNs in this implementation employ the 32-bit single-precision floating-point (FP32) format, which serves as PyTorch's default precision. The training process takes place on a computer system featuring an NVIDIA RTX A4000 GPU and an Intel Core i9-14900K processor.

Table 1. Hyperparameters for PINNs: Tuning ranges and selected values.

Hyperparameter	Range	Chosen Value
No. of Layers	3-9	5
No. of Neurons	32-186	64
Activation function	ReLU, SiLU, Sigmoid, Tanh	Tanh
Collocation points	PDE: 1500-3000	2500
	BC: 1200-2400	2000
	IC: 600-1200	1000
Optimiser	SGD, Adam, L-BFGS-B, Adagrad	Adam
Optimiser Learing rate	Decays quadratically over epochs, based on the following scheduler: $\varphi_0 = 10^{-2}, \varphi_1 = 0.70\varphi_0$ $\varphi_i = \left(1-\left(\frac{i}{N}\right)^2\right)\varphi_0 + \left(\frac{i}{N}\right)^2 \varphi_1, \quad i = 0, \dots, N$ N being the number of epochs.	
loss weighting factors	An adaptive weighting strategy is used [18]: $\alpha_f = \frac{\mathcal{L}_f}{\mathcal{L}_{min}}, \alpha_i = \frac{\mathcal{L}_i}{\mathcal{L}_{min}}, \alpha_b = \frac{\mathcal{L}_b}{\mathcal{L}_{min}}, \alpha_d = \frac{\mathcal{L}_f}{\mathcal{L}_{min}},$ $\mathcal{L}_{min}$ *is the minimum among* $\mathcal{L}_f, \mathcal{L}_i, \mathcal{L}_b, \mathcal{L}_d$	
Iterations (Epochs)	10,000-100,000	60,000

3 CASE STUDIES

The Bascule Bridge in Exeter, as shown in Figure 2, is investigated in this study. The bridge features a 17.28-metre simply supported lifting span, a 6.7-metre-wide carriageway, and a 2-metre-wide footway along the parapet. Its structural configuration includes two longitudinal girders, 17 cross beams, and a composite aluminium deck. This research focuses on the east girder, where sensors measure vertical stiffness. Since the girders are pinned at both ends, each can be approximated as a simply supported beam. The cross beams, bolted to the vertical web stiffeners, primarily transfer shear forces and provide lateral resistance, contributing little to the girders' vertical stiffness. Parameters in Equation (4), outlined in Table 3, are based on section properties derived from structural drawings supplied by Exeter City Council, Devon, UK, as described in [21].

Table 2. PINNs Pseudo-code for the identification of structural parameters in a beam under moving loads.

Algorithm

Input: Spatial and temporal collocation points and measured data (x_i, t_i).

Output: Optimal parameters θ^* and estimated unknown parameters γ^* : $\Theta^* = \{\theta^*, \gamma^*\}$.

Initialisation:

1: Initialise neural network architecture parameters θ
2: Generate random dataset for N_f, N_i *and* N_b
3: Prepare the measured data points N_d
4: Define the total loss functions: $\mathcal{L}(\theta) = \alpha_f \mathcal{L}_f(\theta) + \alpha_i \mathcal{L}_i(\theta) + \alpha_b \mathcal{L}_b(\theta) + \alpha_d \mathcal{L}_d(\theta)$
5: Initialise the loss weights: $\alpha_f, \alpha_i, \alpha_b\ \alpha_d$
6: Initialise the structural parameters γ

Training:

7: Set the optimiser: Adam.
8: Set N: Define the maximum number of training iterations
9: **WHILE** converge not reached **DO**
9.1: **FOR** $k = 1$ to N **DO**
9.2: **FOR** each batch of points in N_f, N_i, N_b, N_d **DO**
9.3: Compute loss function:
- $\mathcal{L}_f(\Theta)$: PDE residual at N_f
- $\mathcal{L}_i(\theta)$: Initial condition at N_i
- $\mathcal{L}_b(\theta)$: Boundary condition at N_b
- $\mathcal{L}_d(\theta)$: Measured data mismatch at N_d

9.4: Update loss function weights $\alpha_f, \alpha_i, \alpha_b\ \alpha_d$
9.5: **END FOR**
9.6: Compute the total loss $\mathcal{L}(\Theta)$
9.7: Update parameters $\Theta = \{\theta, \gamma\}$ using Adam optimiser based on the gradient of $\mathcal{L}(\Theta)$
9.10: **END FOR**
9.11: **END WHILE**
10: Save the trained model: Store the optimised parameters $\Theta^* = \{\theta^*, \gamma^*\}$
11: Deploy the model: Use the trained PINNs to predict the structural response.

Table 3. Dimensions and properties of the Bascule bridge main girder (symmetrical I-section).

Dimensions	Value	Properties	Value
Length	17.28 m	Elastic Modulus (E1)	205 GPa
Depth	926.60 mm	Moment inertia (I)	50.40×10^8 mm^4
Flange width	307.70 mm	Mass per unit length (μ)	289 kg/m
Flange thickness	32 mm		
Web thickness	19.50 mm		

DOI: 10.3217/978-3-99161-057-1-045

Figure 2. Bascule bridge, constructed in 1972, carrying the A379 over the Exeter Canal in Devon, England, UK.

3.1 Case Study 1

In this case study, PINNs are trained to model the effects of a 37 kN axle load, representing the front axle of a truck. This axle load magnitude corresponds to those used in a controlled loading test previously conducted on the Bascule Bridge, as detailed in [22]. The truck's total load consisted of front axles weighing 67.20 kN and rear axles weighing 89.80 kN each. Axle spacing was 2.00 metres between the first and second axles, 3.10 metres between the second and third, and 1.40 metres between the third and fourth. During the test, the truck crossed the bridge in 3.342 seconds, travelling in the lane closest to the east girder, which was estimated to bear approximately 55% of the total load. For the time configuration, the front axle is estimated to take a total of 2.43 seconds to cross the bridge. This timing reflects the sequence and delay with which each of the four axles engages with the bridge as observed during the field tests. Based on these observations, the first front axle enters the bridge at t = 0, and the fourth rear axle exits at t = 3.342 seconds. The truck's speed is calculated using the span length of the bridge (17.28 m), the distance between the first and fourth axles (6.50 m), and the total crossing time of 3.342 seconds. This yields a vehicle speed of 7.115 m/s. Using this speed and the known axle spacings, the times at which the first front axle entered and exited the bridge are determined.

3.2 Synthetic data under varying signal-to-noise ratios

In this case study, the objective is to estimate the bridge's main girder flexural stiffness (EI), deflection, and its derivatives. This requires generating training data for the PINN model, as described in Section 2. To achieve this, we developed a finite element (FE) model comprising 10 beam elements in ANSYS APDL [23]. The model was solved using a transient analysis with 1,000-time steps, providing the synthetic data needed for training. A total of 100 points were randomly selected across the spatial and temporal domains to represent the deflection solution. However, SHM data often contain noise due to environmental and operational factors, such as temperature fluctuations, variable live loads, and sensor inaccuracies. To reflect these real-world conditions, we introduced white Gaussian noise into the simulated data. Incorporating this noise increases the fidelity of the synthetic data, making it more representative of the challenges encountered in real-world SHM data analysis. In addition to estimating the girder's stiffness and deflection, another objective of this study was to assess the adaptability and performance of PINNs when trained with and without noise.

Various noise levels are evaluated by considering different signal-to-noise ratios (SNRs). White Gaussian noise is added to the deflection data generated by the finite element (FE) model, as defined by Equation (17):

$$u_{noise} = u + \mathcal{N}(0, \sigma^2) \tag{17}$$

Where u is the true deflection predicted by the FE model, and $\mathcal{N}(0, \sigma^2)$ represents white Gaussian noise with a mean of zero and a standard deviation σ determined by the target SNR. The SNR in decibels (dB), is given by Equation (18):

$$SNR(dB) = 10\, log_{10}\left(\frac{Signal\ Power}{Noise\ Power}\right) \tag{18}$$

In this study, four SNR levels—10, 20, 30, and 40 dB—are considered and are illustrated in Figure 3. A lower SNR, such as 10 dB, indicates significant noise levels, making the prediction task more challenging. In contrast, a higher SNR, like 40 dB, suggests that noise has only a minor effect.

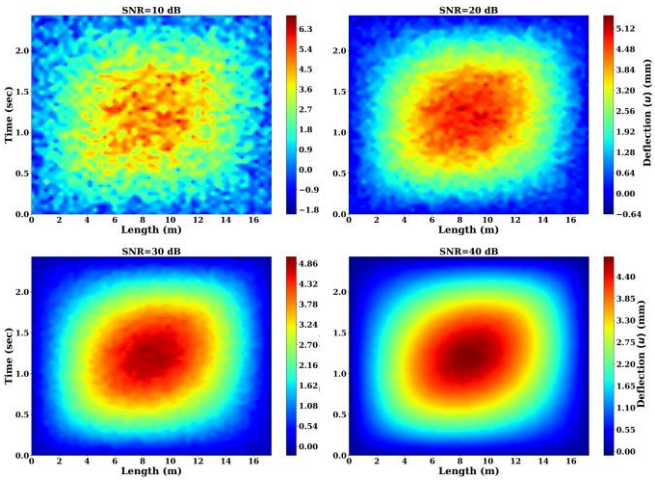


Figure 3. Influence of signal-to-noise ratio (SNR) on deflection data quality.

3.3 Results and discussion of Case Study 1

The results presented in Tables 4 and 5 highlight the performance of the PINNs in predicting the flexural rigidity (EI) and internal forces (deflection and moment) of a Bascule bridge girder under various signal-to-noise ratio (SNR) conditions. The analysis assesses prediction accuracy using relative error percentages (Re%) for EI, as well as Root Mean Square Error (RMSE), Coefficient of Variation (CV%), and Normalised Mean Bias (NMB%) metrics for deflection (u) and moment (Mx) at different time instances. Table 4 indicates that PINNs can provide accurate estimates of EI even in noisy conditions. When using clean data, the relative error is only 0.24%. As noise increases, the error rises slightly, reaching 0.96% at 10dB. However, the PINNs still demonstrate strong robustness, with errors staying below 1% even under the lowest SNR conditions. For higher SNR levels, such as 30dB and 40dB, the errors drop

DOI: 10.3217/978-3-99161-057-1-045

to 0.34% and 0.29%, respectively. This trend confirms that the PINNs can effectively handle varying noise levels, maintaining a high degree of accuracy in estimating the bridge girder's flexural rigidity.

Table 5 examines deflection and moment predictions at different time instances under varying SNR conditions. As expected, lower SNR values lead to higher RMSE, CV, and NMB percentages, reflecting the impact of noise on prediction accuracy. For instance, at 10dB, deflection RMSE ranges from 0.161 mm to 0.178 mm, and CV percentages are as high as 8.26%. In contrast, under a better condition (30dB and 40dB), these values are significantly reduced, with RMSE as low as 0.036 mm and CV percentages dropping to around 1.45%–2.05%. The moment prediction metrics follow a similar pattern, showing that as SNR improves, RMSE and CV percentages consistently decrease. The results of this case study demonstrate that PINNs can accurately predict both flexural rigidity and internal forces in a bridge girder, showing robustness against noise and the potential for practical application in SHM tasks.

Furthermore, Figure 4 illustrates the ability of the PINN trained with data of 40 dB SNR to accurately predict both the beam's deflection and its internal forces, including moments. The data presented in the figure corresponds to a specific time instance, t = 1.22 seconds. Although this example focuses on a single time point for demonstration purposes, the PINN methodology can similarly generate deflection and its derivatives at any chosen moment.

The results displayed in the figure highlight an agreement between the PINN predictions and the FE results. The PINN accurately reproduces the deflection and moment profiles along the beam, aligning closely with the established FE data. This consistent match provides further validation of the PINN's reliability and effectiveness.

Table 4. PINNs predictions of flexural rigidity (EI) for a Bascule bridge girder across varying SNR.

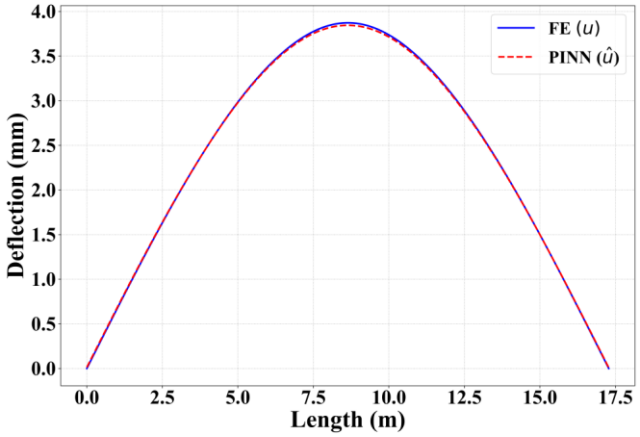
SNR	EI Actual ($N.m^2$)	EI predicted ($N.m^2$)	Re%
Clean	1,033,610	1,036,091	0.24
10dB	===	1,043,533	0.96
20dB	===	1,041,879	0.80
30dB	===	1,037,124	0.34
40dB	===	1,036,607	0.29

3.4 Case Study 2

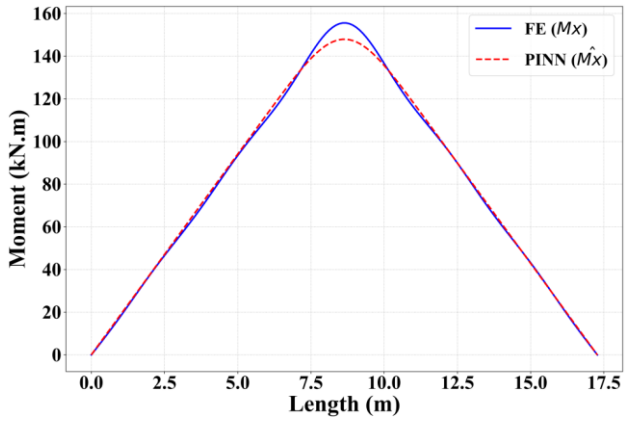
In this case study, a PINN model is used to predict the flexural rigidity and structural behaviour of a girder, informed by strain measurements from sensors installed on the bridge's main girder during a truck crossing. The objective is twofold: to examine the PINN's ability to represent real-world structural behaviour and to assess how measurement data influences its accuracy and generalisation. Four weldable strain gauges were installed, two on the top flange (S1 and S2) and two on the bottom flange (S3 and S4), to measure bending strains along the girder's longitudinal axis as shown in Figure 5. These sensors, featuring a 5.84 mm active grid length and a resistance of 120 **Ω**, were deployed and documented as detailed in [12]. The strain data were collected at 2000 Hz, then baseline-adjusted and smoothed. Figure 6 illustrates the strain time histories recorded by sensors S1 to S4, confirming that the truck took approximately 3.342 seconds to cross, consistent with Case Study 1.[22].

Table 5. PINNs predictions metric errors of deflection and moment for a Bascule bridge girder across varying SNR and different time instances.

SNR	Time (Sec)	Deflection (u)			Moment (Mx)		
		RMSE (mm)	CV %	NBM %	RMSE (kN.m)	CV %	NBM %
Clean	0.60	0.013	0.62	-0.02	3.06	4.07	-0.89
	1.22	0.017	0.57	-0.27	2.94	2.94	-0.18
	1.82	0.014	0.67	-0.32	3.19	4.30	0.37
10dB	0.60	0.161	7.51	7.52	2.75	3.66	-0.13
	1.22	0.178	5.91	5.90	2.90	2.91	0.56
	1.82	0.175	8.26	8.24	3.09	4.16	0.99
20dB	0.60	0.055	2.56	2.13	3.83	5.10	-2.35
	1.22	0.050	1.67	1.09	3.50	3.51	-1.13
	1.82	0.038	1.81	1.51	3.42	4.62	-1.12
30dB	0.60	0.096	4.49	4.44	3.16	4.21	-1.00
	1.22	0.087	2.91	2.84	3.12	3.12	-0.24
	1.82	0.059	2.80	2.74	3.29	4.44	-0.28
40dB	0.60	0.036	1.67	1.52	2.97	3.96	-0.57
	1.22	0.044	1.45	1.29	2.88	2.89	0.25
	1.82	0.044	2.05	1.98	3.29	4.43	0.81



(a) Beam deflection (*u, mm)*



(b) Moment distribution (M_x, *kN.m*)

Figure 4: Case Study 1 – comparing PINNs predictions to finite element (FE) results: (a) beam deflection and (b) moment distribution at t = 1.22 seconds.

DOI: 10.3217/978-3-99161-057-1-045

Unlike Case Study 1, which trained the PINN on a single axle load, this case study models the effect of the entire truck. The truck is represented as four axle loads, spaced at known intervals. The combined load function is approximated as a sum of Dirac delta functions, δ_j, which are expressed in Equations (19) and (20). The load (P) due to all axles is formulated as:

$$P = \sum_{j=1}^{4} p_j \cdot \delta_j(t_j, v, L) \tag{19}$$

Where $\delta_j(t_j, v, L)$ is defined as:

$$\delta_j(t_j, v, L) = \frac{1}{\beta\sqrt{\pi}} e^{-\frac{(x-v(t-t_j))^2}{\beta^2}} \tag{20}$$

The parameters are: L, the beam length; v, the vehicle speed; p_j, the load for the *j-th* axle; β, the regularisation parameter of the Gaussian approximation for δ_j; and t_j, the time lag for the *j-th* axle relative to the first. Incorporating this Dirac delta approximation into the governing PDE yields Equation (21):

$$EI\frac{\partial^4 u(x,t)}{\partial x^4} + \mu\frac{\partial^2 u(x,t)}{\partial t^2} = \sum_{j=1}^{4} p_j \cdot \delta_j(t_j, v, L) \tag{21}$$

In addition, the PINN training process includes a data loss term, $\mathcal{L}_D$ as defined in Equation (15), which quantifies the discrepancy between sensor measurements and the PINN's predicted strains. The strain ε is linked to the beam's curvature $\frac{d^2u(x,t)}{dx^2}$, as shown in Equation (22):

$$\varepsilon = -y\frac{d^2u(x,t)}{dx^2} \tag{22}$$

Here, y is the distance from the neutral axis. This relationship allows $\mathcal{L}_D$ to be computed by comparing measured and predicted strains. The formulation of $\mathcal{L}_D$ is given in Equation (23):

$$\mathcal{L}_d(\theta) = \frac{1}{N_d}\sum_{i=1}^{N_d}\left\| -y\frac{d^2u(x_i,t_i;\,\theta)}{dx^2} - \varepsilon(x_i,t_i;\theta)\right\|^2 \tag{23}$$

$\mathcal{L}_D$ is then incorporated into the overall loss function (Equation 10) to guide the PINN's training and improve its alignment with the observed data.

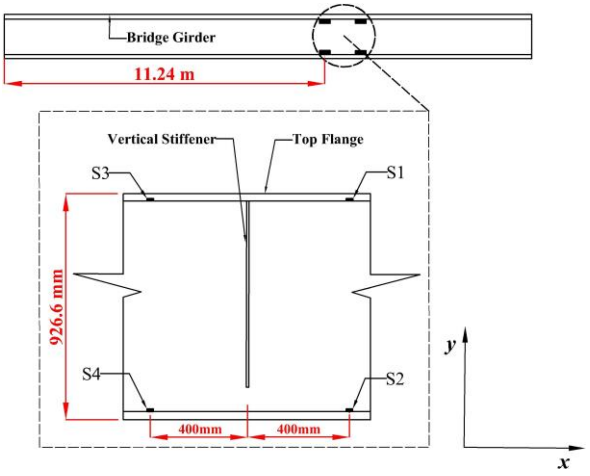


Figure 5. Illustration depicting the positions of weldable strain gauges on the main girder (side view) of a bascule bridge.

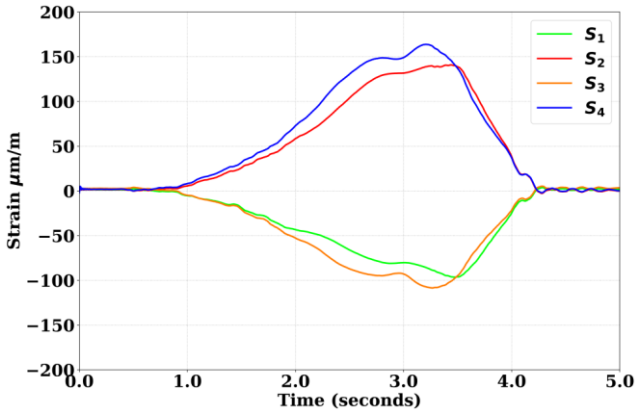


Figure 6. Raw strain measurements from sensors S1 to S4 on the main girder of a bascule bridge as a truck crossed.

3.5 Results and discussion of Case Study 2

The results presented in Table 6 highlight the impact of incorporating real-world strain measurements on the accuracy of PINN predictions for the bridge girder's flexural rigidity (EI). When trained without sensor data, the PINN predicted an EI value that deviated from the actual parameter by over 14%, underscoring the limitations of relying solely on physics-informed constraints. In contrast, the inclusion of strain data from only one sensor S2 reduced the relative error to approximately 1.6%, demonstrating a substantial improvement in predictive performance. This outcome suggests that integrating field measurements enables the PINN to capture the structural behaviour more effectively, enhancing its generalisation and reliability. The ability of PINNs to leverage sparse sensor data offers advantages for SHM applications, providing engineers with more precise estimates of key structural parameters. Additionally, Figure 7 demonstrates the predictive efficacy of the PINN trained with measured data. The model was trained utilising strain data from sensor S2, with the sensor's location and strain time profile shown in Figures 5 and 6, respectively. The displayed results indicate the model's prediction at the position of sensor S4. Accordingly, the results underline the potential for PINNs, when combined with real-world data, to serve as robust tools for both assessing current structural performance.

Table 6. PINN predictions of flexural rigidity (EI) for a Bascule bridge girder with and without sensor Data.

Strain data	EI Actual ($N.m^2$)	EI predicted ($N.m^2$)	Re%
No	1,033,610	1,178,315	14.16
Yes	===	1,050,458	1.63

DOI: 10.3217/978-3-99161-057-1-045

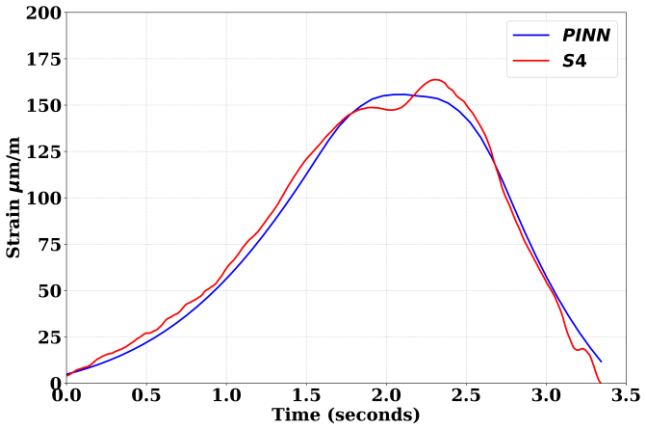


Figure 7. Comparison of strain time history predicted by PINN and measurements from sensor S4.

4 CONCLUSIONS

This study demonstrates the application of a physics-informed neural network (PINN) framework to accurately predict both structural parameters — flexural rigidity (EI) — and structural responses (e.g., deflections and moments) of a bridge girder under a moving load. The investigation is based on two case studies that evaluate the PINN's performance, accuracy, and generalisability, offering valuable insights into its practical application for SHM. In Case Study 1, the PINN was trained under varying levels of synthetic noise to assess its robustness. In Case Study 2, the PINN's ability to integrate real-world strain measurements was explored. Using field measurements, the model achieved significant improvement in prediction accuracy.

The following points summarise the key outcomes from this study:

1. The PINN maintained reasonable accuracy even at low signal-to-noise ratios (SNRs), with predictions becoming increasingly reliable as noise levels decreased. The study revealed that at higher SNRs, the PINN effectively captured the bridge's flexural rigidity, achieving relative errors below 1% and confirming its capacity for precise parameter estimation.
2. In Case Study 2, the PINN trained without sensor data overestimated the girder's flexural rigidity. Incorporating field measurements reduced the relative error from 14% to approximately 1.6%. This demonstrates the value of integrating field data into the PINN framework, enabling more accurate reflection of real-world structural behaviour.
3. The study showed that even a limited number of strain sensors could provide sufficient constraints to enhance the PINN's generalisation and performance, reinforcing its practical utility.
4. The findings show that integrating field measurements not only enhances the accuracy of the flexural rigidity estimation but also allows the model to reflect actual structural behaviour. This demonstrates the PINNs' potential as a practical tool for SHM.

Overall, this research illustrates a novel application of PINNs and contributes to the existing body of knowledge in structural health monitoring (SHM) by introducing a PINN-based approach for estimating the structural parameters of bridge girders. Future work will focus on extending the use of PINNs to damage detection scenarios and integrating multiple PINNs to model more complex structural systems comprising assemblies of discrete elements.

DATA AVAILABILITY

Data will be made available on request.

ACKNOWLEDGMENTS

The first author would like to acknowledge the financial support of The Higher Committee for Education Development in Iraq (HCED) scholarship reference D-14-2968.

REFERENCES

[1] E. Figueiredo and J. M. W. Brownjohn, "Three Decades of Statistical Pattern Recognition Paradigm for SHM of Bridges," Struct Health Monit, vol. 21, no. 6, pp. 3018–3054, 2022, doi: 10.1177/14759217221075241.

[2] R. Chacón, J. R. Casas, C. Ramonell, H. Posada, I. Stipanovic, and S. Škarić, "Requirements and Challenges for Infusion of SHM Systems Within Digital Twin Platforms," Structure and Infrastructure Engineering, pp. 1–17, 2023, doi: 10.1080/15732479.2023.2225486.

[3] T. Liu and H. Meidani, "Physics-Informed Neural Networks for System Identification of Structural Systems with a Multiphysics Damping Model," J Eng Mech, vol. 149, no. 10, Oct. 2023, doi: 10.1061/jenmdt.emeng-7060.

[4] 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," J Comput Phys, vol. 378, pp. 686–707, 2019, doi: 10.1016/j.jcp.2018.10.045.

[5] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific Machine Learning Through Physics–Informed Neural Networks: Where we are and What's Next," J Sci Comput, vol. 92, no. 3, pp. 1–62, 2022, doi: 10.1007/s10915-022-01939-z.

[6] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Inferring Solutions of Differential Equations Using Noisy Multi-Fidelity Data," J Comput Phys, vol. 335, pp. 736–746, 2017, doi: 10.1016/j.jcp.2017.01.060.

[7] Z. Mao, A. D. Jagtap, and G. E. Karniadakis, "Physics-Informed Neural Networks for High-Speed Flows," Comput Methods Appl Mech Eng, vol. 360, no. 112789, 2020, doi: 10.1016/j.cma.2019.112789.

[8] M. Rasht-Behesht, C. Huber, K. Shukla, and G. E. Karniadakis, "Physics-Informed Neural Networks (PINNs) for Wave Propagation and Full Waveform Inversions," J Geophys Res Solid Earth, vol. 127, no. 5, May 2022, doi: 10.1029/2021JB023120.

[9] Y. Martinez, L. Rojas, A. Peña, M. Valenzuela, and J. Garcia, "Physics-Informed Neural Networks for the Structural Analysis and Monitoring of Railway Bridges: A Systematic Review," Mathematics, vol. 13, no. 10, p. 1571, May 2025, doi: 10.3390/math13101571.

[10] A. I. F. Al-Adly and P. Kripakaran, "Physics-informed neural networks for structural health monitoring: a case study for Kirchhoff–Love plates," Data-Centric Engineering, vol. 5, p. e6, 2024, doi: 10.1017/dce.2024.4.

[11] I. J. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.

[12] S. P. Timoshenko, History of Strength of Materials. New York: McGraw-Hill, 1953.

[13] J. M. Gere and S. Timoshenko, Mechanics of Materials. in Mechanics of Materials. PWS Publishing Company, 1997.

[14] Ladislav Fryba, Vibration of Solids and Structure under Moving Loads, vol. 3rd edition. Thomas Telford (ICE Publishing), 1999.

[15] John M. Biggs, Introduction to Structural Dynamics. New York: McGraw- Hill , 1964.

[16] L. Meirovitch, Analytical Methods in Vibrations. United State of America: The Macmillan Company, 1967.

DOI: 10.3217/978-3-99161-057-1-045

[17] S. A. Eftekhari, “A differential quadrature procedure with regularization of the Dirac-delta function for numerical solution of moving load problem,” Latin American Journal of Solids and Structures, vol. 12, no. 7, pp. 1241–1265, 2015, doi: 10.1590/1679-78251417.

[18] H. M. Zhang, X. Shao, Z. F. Zhang, and M. Y. He, “E-PINN: extended physics informed neural network for the forward and inverse problems of high-order nonlinear integro-differential equations,” Int J Comput Math, vol. 101, no. 7, pp. 732–749, 2024, doi: 10.1080/00207160.2024.2374820.

[19] G. Van Rossum and F. L. Drake, “Python Language Reference, version 3.11,” 2022, Python Software Foundation.

[20] J. Ansel et al., “PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation,” in Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2, in ASPLOS ’24. New York, NY, USA: Association for Computing Machinery, 2024, pp. 929–947. doi: 10.1145/3620665.3640366.

[21] J. Kwad, “Estimating fatigue life of steel bridges using continuous response monitoring : Methodology and applications,” Doctor of Philosophy in Engineering, University of Exeter, 2018.

[22] J. Kwad and P. Kripakaran, “Hybrid approach combining modelling and measurement for fatigue damage estimation of welded connections in bridges,” Structure and Infrastructure Engineering, vol. 17, no. 1, pp. 20–33, 2021, doi: 10.1080/15732479.2020.1730407.

[23] Ansys, “ANSYS Worckbench, Release number 2023 R1, Help System, Theory Guide, ANSYS, Inc.,” 2023, ANSYS, Inc., Canonsburg, USA: Release number 2023 R1.