Chapter 1
Modelling a Multi-Degree-of-Freedom Beam using Lagrangian Neural Networks

Alan Xavier and Ludovic Renson

Abstract Accurately modelling the dynamic behaviour of nonlinear structures is challenging due to the wide range of potential nonlinearities and dynamic phenomena that they can exhibit. Physics-guided machine learning (PGML) has emerged as an attractive way to combine prior knowledge with data to solve a wide range of complex, nonlinear problems in science and engineering. Lagrangian Neural Networks (LNNs) are a particular PGML approach that models nonlinear systems' Lagrangian functions using artificial neural networks (NNs). The Euler-Lagrange equation is then reconstructed through automatic differentiation (AD) to derive the equations of motion, enforcing physical consistency during training.

Previously, we used LNNs to model the nonlinear vibrations of a single-degree-of-freedom (SDOF) Duffing oscillator. We extend our earlier works to address systems with multiple-degrees-of-freedoms (MDOF) and demonstrate the method on experimental data collected on a cantilever beam featuring complex response features, such as internal resonance, isola, and Neimark-Sacker bifurcations. We analyse the physical consistency of the trained model and interpret the identified stiffness and damping nonlinearities from the partial derivatives of the potential energy and dissipation functions. The trained LNNs are then used to trace frequency response and bifurcation curves, which can be directly compared to curves measured with control-based continuation.

Keywords Physics-guided machine learning · Lagrangian neural networks · Control-based continuation

Introduction

Physics-guided machine learning (PGML) integrates physics principles with data-driven models to achieve improved accuracy and generalization compared to either approach alone. Within this framework, Cranmer et al. [1] introduced Lagrangian Neural Networks (LNNs), where the system dynamics are derived from a neural-network approximation of the Lagrangian, $\mathscr{L}_\theta(q, \dot{q})$.

For structural dynamics, however, purely conservative formulations are insufficient, since real systems experience both external excitation and dissipation. To account for these effects, we extend the standard Euler–Lagrange equations by including a dissipation function $\mathscr{D}(\dot{q})$ and an external forcing term $\mathscr{F}(t)$ [2, 3]:

$$\frac{d}{dt}\left(\frac{\partial \mathscr{L}}{\partial \dot{q}_j}\right) - \frac{\partial \mathscr{L}}{\partial q_j} + \frac{\partial \mathscr{D}}{\partial \dot{q}_j} - \mathscr{F}(t) = 0. \tag{1}$$

This yields the estimated accelerations:

$$\hat{\ddot{q}} = \left(\nabla_{\dot{q}} \nabla_{\dot{q}}^T \mathscr{L}_\theta\right)^{-1} \left[\nabla_q \mathscr{L}_\theta - \left(\nabla_q \nabla_{\dot{q}}^T \mathscr{L}_\theta\right) \dot{q} - \nabla * \dot{q} \mathscr{D}_\theta + \mathscr{F}(t)\right], \tag{2}$$

where $\mathscr{L}_\theta$ and $\mathscr{D}_\theta$ are neural networks representing the conservative and dissipative dynamics, respectively.

To ensure physically consistent rest states, we impose $\mathscr{L}(0, 0) = 0$ and $\mathscr{D}(0) = 0$. The networks are trained by mini-

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

Emily Retzlaff et al. (eds.), *Nonlinear Structures & Systems, Vol. 1*, Conference Proceedings
of the Society for Experimental Mechanics Series, https://doi.org/10.1007/978-8-743-81440-5_1

mizing a loss function that balances data fidelity with physics-based constraints:

$$loss = \lambda_0 \cdot |\hat{\ddot{q}} - \ddot{q}| + \sum_{i=1}^{n} \lambda_i \cdot \mathscr{G}_i, \tag{3}$$

where $\ddot{q}$ are measured (or simulated) accelerations and $\mathscr{G}_i$ encodes additional constraints. This allows us to enforce any prior knowledge about the system to improve our network's performance as demonstrated in [3]. This formulation allows the LNN to learn both the conservative structure and the non-conservative dynamics of the system directly from data.

Application to a Multi-Degree-of-Freedom Beam

To evaluate the LNN approach, we consider a continuous beam, shown schematically in Figure 1 (a). This setup captures the essential features of a continuous structure under external excitation.

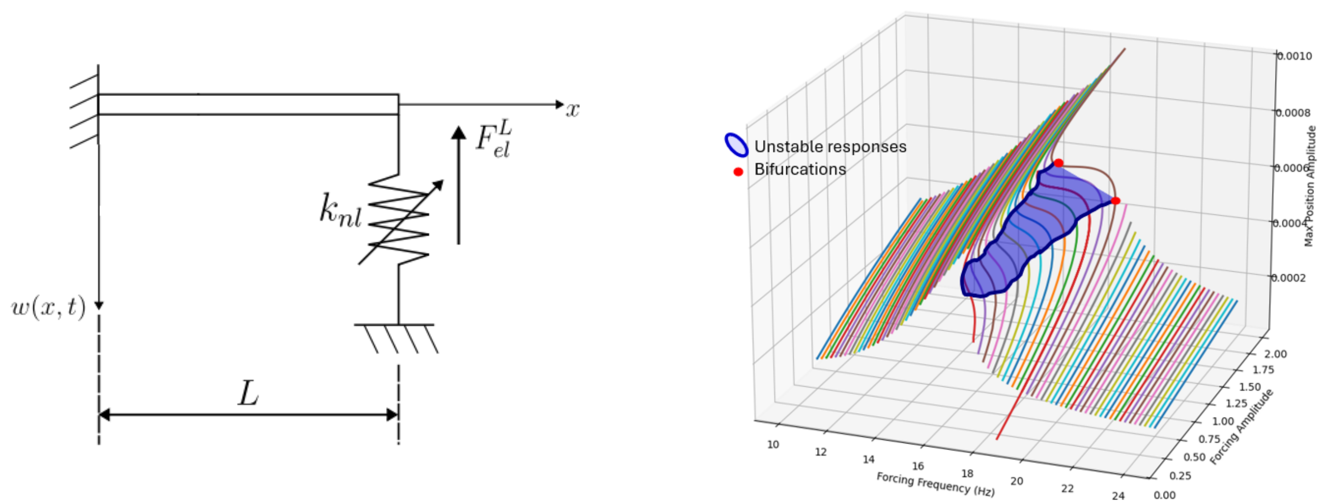

Fig. 1 (a) Picture of the beam system considered and (b) Forced response of the system showing the steady-state periodic responses computed during a series of continuations in forcing amplitude (S-curves). A portion of the dataset includes unstable responses (shaded in blue), highlighting the challenge for the LNN to accurately learn and reproduce such nonlinear dynamics

The transverse displacement of an Euler–Bernoulli beam, $w(x,t)$, is governed by the PDE (4):

$$\rho A \frac{\partial^2 w}{\partial t^2} + c\frac{\partial w}{\partial t} + EI\frac{\partial^4 w}{\partial x^4} = f(x,t) + F_{el}^L, \tag{4}$$

where ρA is the mass per unit length, EI is the flexural rigidity, c is a damping coefficient, $f(x,t)$ is the applied force, and F_{el}^L is a cubic external elastic force at the free end.

Since the beam is a continuous system with an infinite number of vibration modes, we approximate its dynamics by truncating to $N = 2$ dominant modes:

$$w(x,t) \approx \sum_{i=1}^{N} \phi_i(x) q_i(t), \tag{5}$$

where $\phi_i(x)$ are mode shapes and $q_i(t)$ are generalized coordinates. Substituting this expansion into (4) and applying the Galerkin method yields the reduced-order system:

$$M\ddot{q}(t) + C\dot{q}(t) + Kq(t) + f_{nl}(q(t), \dot{q}(t)) = F(t), \tag{6}$$

with M, C, and K being the modal mass, damping, and stiffness matrices, respectively, and f_{nl} accounting for any nonlinearities.

Thus, the beam system is described by equation (6), yielding a multi-degree-of-freedom (MDOF) model suitable for training and testing the LNN.

Simulation Data

The dataset was obtained from time-domain simulations of the beam under harmonic excitation, carried out using numerical continuation. Each simulation produced an S-curve, i.e., a set of steady-state responses measured at a fixed excitation frequency, ω as shown in Figure 1 (b). Frequencies were varied between 10.0 Hz and 24.0 Hz in increments of 0.2 Hz, resulting in 71 simulation runs and a dataset of order $\mathscr{O}(10^5)$ time samples.

The dataset was divided using an $80/20$ split applied within each S-curve rather than across the full collection of curves. This approach ensures that both training and test sets include data from the entire frequency range, while the test set still contains forcing amplitudes not seen during training. The per-S-curve split therefore provides a balanced assessment of the model's ability to generalize to unseen amplitudes within each frequency branch, rather than merely interpolating across branches. By contrast, splitting across full S-curves would exclude certain frequency branches entirely from training, creating a more stringent extrapolation task but at the cost of reduced frequency coverage during training.

As shown in Figure 1 (b), a part of the dataset includes unstable responses (highlighted in blue). This serves as both a challenging test of the LNN's learning capability and an opportunity to assess how well it reproduces such nonlinear dynamics in practice.

Network Training

The LNN architecture comprises three neural networks: $\mathscr{T}_\theta(q, \dot{q})$ for the kinetic energy, $\mathscr{V}_\theta(q, \dot{q})$ for the potential energy, and $\mathscr{D}_\theta(\dot{q})$ for the damping contribution. The kinetic and potential networks each consist of four hidden layers with 64 neurons per layer, while the damping network uses four hidden layers with 32 neurons per layer. All hidden layers employ the $\tanh$ activation function, with linear outputs at the final layer. To improve training stability, a normalization layer was applied. The networks were trained for 200 epochs using the Adam optimizer with a learning rate of 10^{-4}.

Figure 2 (a) compares predicted accelerations $\hat{\ddot{q}}$ with simulated ground truth $\ddot{q}$ across test trajectories. The LNN reproduces the overall response well, though discrepancies remain in capturing higher-order harmonics.

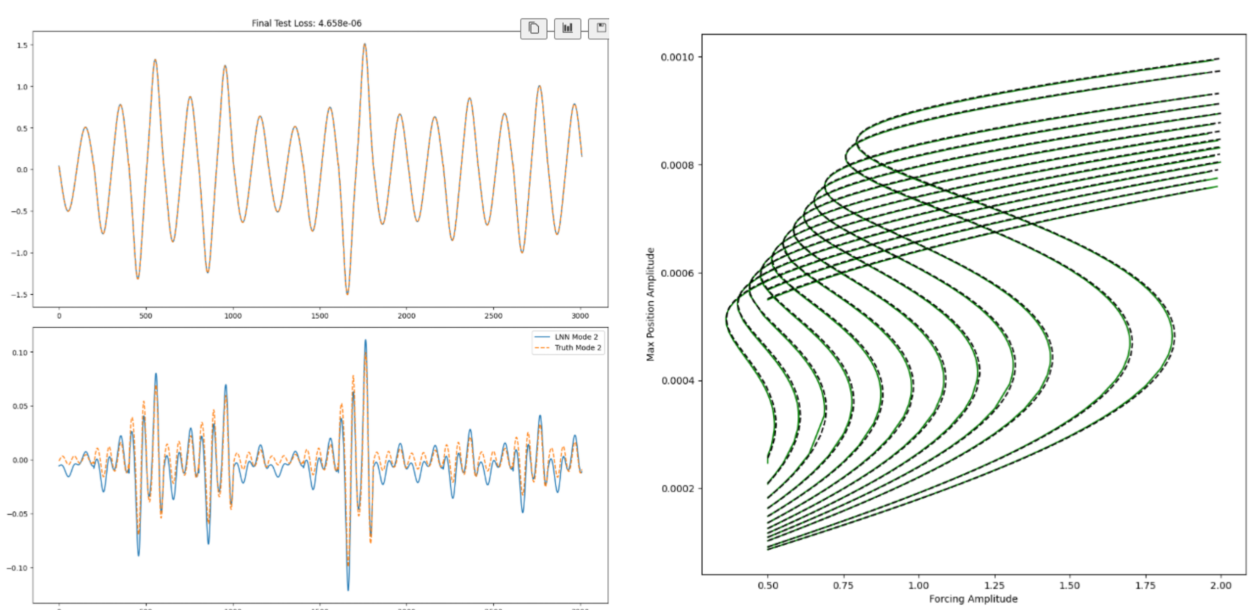

Fig. 2 Comparison of predicted accelerations, $\hat{\ddot{q}}$, from the trained LNN with ground-truth simulations, $\ddot{q}$, across sample test trajectories. (b) Backbone and forced response curves obtained via numerical continuation with the trained LNN (green), compared against the reference beam model (black). Results are shown for excitation frequencies where unstable responses and bifurcations arise (16.2 and 19.0 Hz)

Numerical Continuation

To assess the predictive capability of the trained LNN, we employed it as a surrogate within a numerical continuation scheme. Continuation was performed at fixed excitation frequencies, ω, with the forcing amplitude used as the continuation parameter. At each step, steady-state responses were obtained by integrating the learned equations of motion, and these were tracked along the continuation path.

Figure 2 (b) compares backbone and forced response curves from the LNN with those from the reference beam model, focusing on frequencies where unstable responses or bifurcations occur. These results highlight the extent to which the LNN captures key nonlinear features, including the onset of bifurcations.

References

1. Cranmer, M.D., Greydanus, S., Hoyer, S., Battaglia, P.W., Spergel, D.N., and Ho, S. “Lagrangian neural networks”. *CoRR*, abs/2003.04630 (2020)
2. Meirovitch, L. *Methods of Analytical Dynamics*. Advanced engineering series. Dover Publications (2010)
3. M. J. Szydlowski, C.S. and Renson, L. “Modeling nonlinear structures using physics-guided, machine-learnt models”. In *Proceedings of the 41st IMAC, A Conference and Exposition on Structural Dynamics 2023* (2023)