



Discussing the spectrum of physics-enhanced machine learning: a survey on structural mechanics applications

Marcus Haywood-Alexander¹ , Wei Liu^{2,3}, Kiran Bacsa^{1,3}, Zhilu Lai^{4,5}  and Eleni Chatzi^{1,3} 

¹Department of Civil, Environmental and Geomatic Engineering, ETH Zürich, Zürich, Switzerland

²Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore, Singapore

³Future Resilient Systems, Singapore-ETH Centre, Singapore, Singapore

⁴Internet of Things Thrust, HKUST(GZ), Guangzhou, China

⁵Department of Civil and Environmental Engineering, HKUST, Hong Kong, China

Corresponding author: Marcus Haywood-Alexander; Email: mhaywood@ethz.ch

Received: 18 October 2023; **Revised:** 22 April 2024; **Accepted:** 11 July 2024


Keywords: data-driven; hybrid learning; physics-based; physics-encoded; physics enhanced; physics-guided; structural mechanics

Abstract

The intersection of physics and machine learning has given rise to the *physics-enhanced* machine learning (*PEML*) paradigm, aiming to improve the capabilities and reduce the individual shortcomings of data- or physics-only methods. In this paper, the spectrum of *PEML* methods, expressed across the defining axes of physics and data, is discussed by engaging in a comprehensive exploration of its characteristics, usage, and motivations. In doing so, we present a survey of recent applications and developments of *PEML* techniques, revealing the potency of *PEML* in addressing complex challenges. We further demonstrate the application of select such schemes on the simple working example of a single degree-of-freedom Duffing oscillator, which allows to highlight the individual characteristics and motivations of different “genres” of *PEML* approaches. To promote collaboration and transparency, and to provide practical examples for the reader, the code generating these working examples is provided alongside this paper. As a foundational contribution, this paper underscores the significance of *PEML* in pushing the boundaries of scientific and engineering research, underpinned by the synergy of physical insights and machine learning capabilities.

Impact Statement

This paper discusses methods born from the fusion of physics and machine learning, known as *physics-enhanced* machine learning (*PEML*) schemes. By considering their characteristics, this work clarifies and categorizes *PEML* techniques, aiding researchers and users to targetedly select methods on the basis of specific problem characteristics and requirements. The discussion of *PEML* techniques is framed around a survey of recent applications/developments of *PEML* in the field of structural mechanics. A running example of a Duffing oscillator is used to highlight the traits and potential of diverse *PEML* approaches. Additionally, code is provided to foster transparency and collaboration. The work advocates the pivotal role of *PEML* in advancing computing for engineering through the merger of physics-based knowledge and machine learning capabilities.

 This research article was awarded an Open Materials badge for transparent practices. See the Data Availability Statement for details.

© The Author(s), 2024. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.



1. Introduction

With the increase in both computing power and data availability, machine learning (ML) and deep learning (DL) are in scientific and engineering applications (Reich, 1997; Hey et al., 2020; Zhong et al., 2021; Cuomo et al., 2022). Such methods have shown enormous potential in yielding efficient and accurate estimates over highly complex domains, such as those with high-dimensionality, or ill-posed problem definitions. The use of data-driven methods is wide reaching in science, from fields such as fluid dynamics (Zhang and Duraisamy, 2015), geoscience (Bergen et al., 2019), bioinformatics (Olson et al., 2018), and more (Brunton and Kutz, 2022). Data-driven schemes are particularly suited for the case of monitored systems, where the availability of data is ensured via the measurement of engineering quantities through the use of appropriate sensors (Sohn et al., 2003; Farrar and Worden, 2007; Lynch, 2007).

However, such data-driven models are known to be restricted to the domain of the instance in which the data was collected; that is, they lack generalisability (O'Driscoll et al., 2019; Karniadakis et al., 2021), as a result of a lack of physical connotation. This challenge is often met when dealing with data-driven approaches for environmental and operational normalization (Cross et al., 2011; Avendaño-Valencia et al., 2017); it is impossible, or impractical, to collect data over the full environmental/operational (E/O) envelope (Figueiredo et al., 2011). Particularly to what concerns data gathered from large-scale engineered systems, it is common to meet a scarcity of training samples across a system's comprehensive operational envelope (Sohn, 2007). These variables frequently exhibit intricate and non-stationary patterns changing over time. Consequently, the limited pool of labeled samples available for training or cross-validation can fall short of accurately capturing the intrinsic relationships for scientific discovery tasks, potentially resulting in misleading extrapolations (D'Amico et al., 2019). This scarcity of representative samples sets scientific problems apart from more mainstream concerns like language translation or object recognition, where copious amounts of labeled or unlabeled data have underpinned recent advancements in deep learning (Jordan and Mitchell, 2015; Sharifani and Amini, 2023). Discussions on and examples of the challenge posed by comparatively small datasets in scientific machine learning can be found in Shaikhina et al. (2015), Zhang and Ling (2018), and Dou et al. (2023).

While black box data-driven schemes are often sufficient for delivering an actionable system model, able to act as an estimator or classifier, a common pursuit within the context of mechanics lies in knowledge discovery (Geyer et al., 2021; Naser, 2021; Cuomo et al., 2022). In this case, it is imperative to deliver models that are explainable/interpretable and generalizable (Linardatos et al., 2020). This entails revealing and comprehending the cause-and-effect mechanisms underpinning the workings of a particular engineered system. Consequently, even if a black-box model attains marginally superior accuracy, its inability to unravel the fundamental underlying processes renders it inadequate for furthering downstream scientific applications (Langley et al., 1994). Conversely, an interpretable model rooted in explainable theories is better poised to guard against the learning of spurious data-driven patterns that lack interpretability (Molnar, 2020). This becomes particularly crucial for practices where predictive models are of the essence for risk-based assessment and decision support, such as the domains of structural health monitoring (Farrar and Worden, 2012) and resilience (Shadabfar et al., 2022).

In modeling complex systems, there is a need for a balanced approach that combines physics-based and data-driven models (Pawar et al., 2021). Modern engineering systems, involve complex materials, geometries, and often intricate energy harvesting and vibration mitigation mechanisms, which may be associated with complex mechanics and failure patterns (Duenas-Osorio and Vemuru, 2009; Van der Meer et al., 2012; Kim et al., 2017). This results in behavior that cannot be trivially described purely on the basis of data observations or via common, and often simplified, modeling assumptions. In efficiently modeling such systems, a viable approach is to integrate the aspect of physics, which is linked to forward modeling with the aspect of learning from data (via machine learning tools), which can account for modeling uncertainties and imprecision. This fusion has been referred to via the term "physics-enhanced machine learning (PEML)" (Faroughi et al., 2022), which we also adopt herein. This term is used to denote that, in

some form, prior physics knowledge is embedded to the learner (O’Driscoll et al., 2019; Choudhary et al., 2020; Xiaowei et al., 2021). which typically results in more interpretable models.

In this work, we focus on applications of *PEML* in the domain of structural mechanics; a field that impacts the design, building, monitoring, maintenance, and disuse of critical structures and infrastructures. Some of the greatest impact comes from large-scale infrastructure, such as bridges, wind turbines, and transport systems. However, accurate and robust numerical models of complex structures are non-trivial to establish for tasks such as Digital Twinning (DT) and Structural Health Monitoring (SHM), where both precision and computational efficiency are of the essence (Farrar and Worden, 2012; Yuan et al., 2020). This has motivated the increased adoption of ML or DL approaches for generating models of such structures, overcoming the challenges presented by the complexity. Further to the extended use in the DT and SHM contexts, data-driven approaches have further been adopted for optimizing the design of materials and structures (Guo et al., 2021; Sun et al., 2021). Multi-scale modeling of structures has also benefited from the use of ML approaches, typically via the replacement of computationally costly representative volume element simulations with ML models, such as neural networks (Huang et al., 2020) or support vector regression and random forest regression (Reimann et al., 2019).

In order to contextualize *PEML* for use within the realm of structural mechanics applications, we here employ a characterization that adopts the idea of a spectrum, as opposed to a categorization in purely white, black, and gray box models. This is inspired by the categorization put forth in the recent works of Cross et al. (2022), which discusses the placement of *PEML* methods on a two-dimensional spectrum of physics and data, and Faroughi et al. (2022) which categorizes these schemes based on the implementation of physics within the ML architecture. In the context of the previously used one-dimensional spectrum, the “darker” end of the said spectrum relies more heavily on data, whilst the “lighter” end relies more heavily on the portion of physics that is considered known (Figure 1). One can envision this one-dimensional spectrum lying equivalently along the diagonal from the red (top left), to the blue (bottom right) corners of the two-dimensional spectrum. Under this definition, “off-the-shelf” ML approaches more customarily fit the black end of the spectrum, while purely analytical solutions would sit on white end of the spectrum (Rebillat et al., 2023). Generally, the position along this spectrum is driven by both the amount of data available, and the level of physics constraints that are applied. However, it is important to note, that the inclusion of data is not a requirement for *PEML*. An example of the latter is found in methods such as *physics-informed neural networks* (PINNs) (Raissi et al., 2019), which exploit the capabilities of

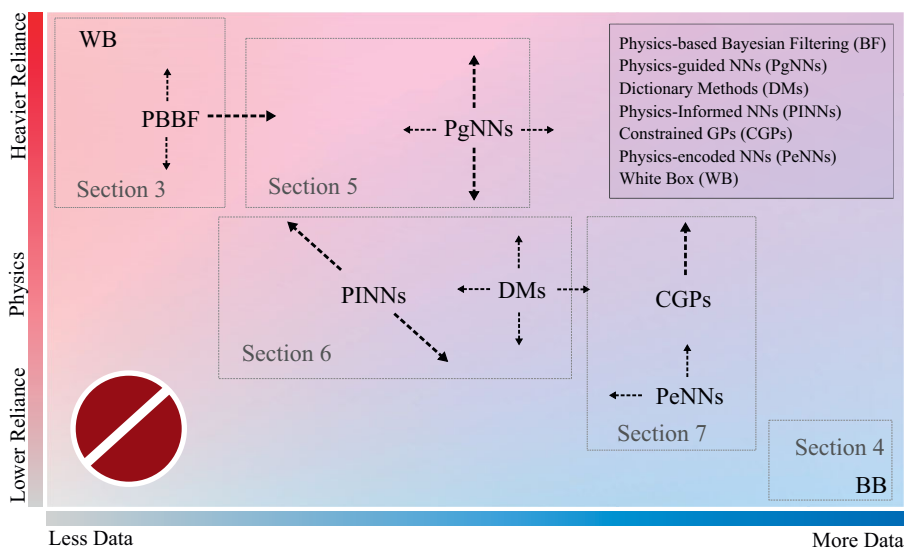


Figure 1. The spectrum of physics-enhanced machine learning (*PEML*) schemes is surveyed in this paper.

ML methods to act as forward modelers, where no observations or measured data are necessarily used for the formulation of the loss function. In these cases, prescribed boundary/initial conditions, physics equations, and system inputs are provided and the ML algorithm is used to “learn” the solutions; one such method is the observation-absent *PINN* (Rezaei et al., 2022).

The reliance on physics can be quantified in terms of the level of strictness of the physics model prescription. The level of strictness refers to the degree to which the prescribed model form incorporates and adheres to the underlying physical principles, and concurrently defines the set of systems which the prescribed model can emulate. For example, when system parameters are assumed known, the physics is more strictly prescribed. Using a solid mechanics example, a strictly prescribed model would correspond to the use of a specified a Finite Element model as the underlying physics structure. This strictness is somewhat relaxed when it is assumed that the model parameters are uncertain and subject to updating (Papadimitriou and Katafygiotis, 2004). An example of a low degree of strictness corresponds to prescribing the system output as a function of the derivative of system inputs with respect to time; such a more loose prescription would be capable of emulating structural dynamics system (Bacsa et al., 2023), as well as further system and problem types, such as heat transfer (Dhadphale et al., 2022), or virus spreading (Núñez et al., 2023). In this work, the vertical axis of the *PEML* spectrum is defined in terms of the reliance on the imposed physics-based model form, which is earlier referred to as the level of strictness. A separate notion to consider, which is not reflected in the included axes, pertains to the level of constraint of the employed *PEML* architecture, which describes the degree to which the learner must adhere to a prescribed model. As an example, residual modeling techniques (Christodoulou and Papadimitriou, 2007) have a relatively low level of constraint, as the solution space is not limited to that which is posited by the physics model. The combination of the strictness in the prescription of a model form and the learner constraints defines the overall *flexibility* of the *PEML* scheme; this refers to its capability to emulate systems of varying types and complexities (Karniadakis et al., 2021).

When selecting the type, or “genre” of *PEML* model, the confidence in the physics that is known a priori guides the selection of the appropriate reliance on physics in the form of the level of strictness of the prescribed model and/or constraints of the learner. Different prior knowledge can be in the form of an appropriate model structure, that is an equivalent MDOF system, or an appropriate finite element model, or it could be that appropriate material/property values are prescribed. The term appropriate here refers to the availability of adequate information on models and parameters, which approximate well the behavior and traits of the true system. If one wishes to delve deeper into such a categorization, the level of knowledge can be appraised in further sub-types, that is, the discrete number of, or confidence in, known material parameters, or the complexity of the model in relation to the real structure.

The remainder of the paper is organized into methods corresponding to different collective areas over the *PEML* spectrum, as indicated in Figure 1. We initiate with a discussion and corresponding examples that are closer to a *white-box* approach in Section 3, where physics-based models of specified form are fused with data via a Bayesian Filtering (BF) approach. This is followed by Section 4, which shows a brief survey of solely data-driven methods, which embed no prior knowledge on the underlying physics. After introducing instances of methods that are situated near the extreme corners of the spectrum (black- and white-box), the main motivation of the paper, namely the overview of *PEML* schemes initiates. The breakdown into the subsections of *PEML* techniques is driven by a combination of the reliance on the prescription of the physics model form and the method of physics embedding, these are broken down below and their naming conventions are explained.

Firstly, Section 5 surveys and discusses *physics-guided* machine learning (PGML) techniques, in which the physics model prescriptions are embedded as proposed solutions, and act in parallel to the data-driven learner in the full *PEML* model architecture. *PGML* schemes steer the learner toward a desired solution by prescribing models with a relatively large degree of strictness, therefore neighboring the similarly strict construct of BF methods. Physics-guided methods often benefit from a reduced data requirement since the physics embedding allows for estimation in the absence of dense observations

from the system. However, depending on the formulation, or the type of method used, such schemes can still suffer from data-sparsity. In Section 6, *physics-informed* methods are presented, which correspond to a heavier reliance on data, while still retaining a moderate reliance on the prescribed physics. These schemes are so named as physics is embedded as prior information, from which an objective or loss function is constructed, which the learner is prompted to follow. Compared to physics-guided methods, in physics-informed schemes, the physics is embedded in a less constrained manner, that is it is weakly imposed. In this sense, such schemes are often formed by way of minimizing a loss or objective function, which vanishes when all the imposed physics are satisfied. The survey portion of the paper concludes with a discussion on *physics-encoded* learners in Section 7. Physics-encoded methods embed the imposed physics directly within the architecture of the learner, via selection of operators, kernels, or transforms. As a result, these methods are often less reliant on the model form (e.g., they may simply impose derivatives), but they are highly constrained in the fact that this imposed model is always adhered to. The position of *PgNNs* and *PINNs* compared to *constrained Gaussian Processes* (CGPs) is less indicative of the higher requirement of CGPs for data, but more indicative of the lower requirement of *PgNNs* and *PINNs* for data, as these methods are capable of proposing viable solutions with fewer data. However, this is dependent on the physics and ML model form, thus arrows are included to indicate their mobility on the spectrum.

Throughout the paper, a working example of a single-degree-of-freedom Duffing oscillator, the details of which are offered in Section 2, is used to demonstrate the methods surveyed. As previously mentioned, the code used to generate the fundamental versions of these methods is provided alongside this paper in a GitHub repository.¹ This code is written in Python and primarily built with the freely available Pytorch package (Paszke et al., 2019).

2. A working example

Aiding the survey and discussion in each aspect of *PEML*, an example of a dynamic system will be used throughout the paper to provide a tangible example for the reader. A variety of *PEML* methods will be applied to the presented model, the aim of which is not to showcase any particularly novel applications of the methods, but to help illustrate and discuss the characteristics of the *PEML* variants for a simple example, while highlighting emerging schemes and their placement across the spectrum of Figure 1. To this end, we employ a single-degree-of-freedom (SDOF) Duffing oscillator, shown in Figure 2a, as a running example. The equation of motion of this oscillator is defined as,

$$m\ddot{u}(t) + c\dot{u}(t) + ku(t) + k_3u^3(t) = f(t), \quad (1)$$

where the values for the physical parameters m , c , k , and k_3 are 10 kg, 1Ns/m, 15 N/m, and 100N/m³, respectively. To be consistent with problem formulations in this paper, this is defined in state-space form as follows:

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{A}_n u^3 + \mathbf{B}f, \quad (2)$$

where $\mathbf{z} = \{u, \dot{u}\}^T$ is the system state, and the state matrices are,

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -m^{-1}k & -m^{-1}c \end{bmatrix}, \quad \mathbf{A}_n = \begin{bmatrix} 0 \\ -m^{-1}k_3 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ m^{-1} \end{bmatrix}.$$

For this example, the forcing signal of the system consists of a random-phase multi-sine signal containing frequencies of 0.7, 0.85, 1.6, and 1.8 rad/s. The Duffing oscillator is simulated using a 4th-order Runge–Kutta integration. The forcing and resulting displacement for 1024 samples at an equivalent sample rate of 8.525 Hz is shown in Figure 2b. These data are then used as the ground truth for the examples shown throughout the paper.

¹ Python code and data are available at <https://github.com/ETH-IBK-SMECH/PIDyNN>.

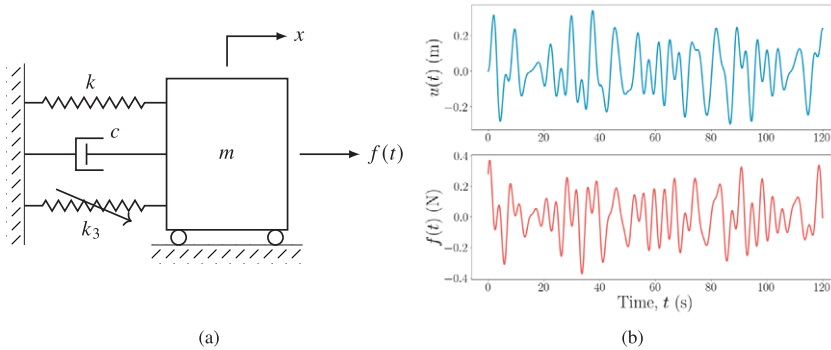


Figure 2. (a) Diagram of the working example used throughout this paper, corresponding to a Duffing Oscillator; instances of the (b) displacement (top) and forcing signal (bottom) produced during simulation.

2.1. A note on data and domains

The interdisciplinary nature of *PEML* can lead to confusion regarding the terms defining the data and domain for the model. To enhance clarity for readers with diverse backgrounds, we provide clarifications on the nomenclature in this paper. Firstly, *data* refers to all measured or known values that are used in the overall architecture/methodology, not exclusively that which is used as inputs to the ML model, or as target observations. This may include, but is not limited to, measured data, system parameters, and scaling information. In the context of the example above, the data encompasses measured values of the state, \mathbf{z}^* , and force, f^* , along with system parameters m, c, k, k_3 (where the asterisk denotes observations of a value). Importantly, the use of the term *observation data* is akin to the classic ML definition of *training data*, which is the scope of data used in traditional learning paradigms which minimize the discrepancy between the model output and some observed target values. This change is employed here as the training stages in many instances of *PEML* demonstrate the learner's ability to make predictions beyond the scope of these observations of target values.

This goal of extending the scope of prediction also prompts a clarification of the term *domain*. The *domain* here is similar to the definition of the domain of a function, representing the set of values passed as input to the model—in this case, the set of time values, t . The domain where *measured* values of the model output are available, is termed the *observation domain*, Ω_o . The overall domain in which the model is *trained*, and predictions can be made, is the *collocation domain*, Ω_c . For example, if one provides measurements of the state for the first third of the signal in Figure 2b but proposes the model to learn (and therefore predict) over the full signal range, the observation domain would be $\Omega_o \in 0 \leq t \leq 40s$ and the collocation domain $\Omega_c \in 0 \leq t \leq 120s$. It is crucial to note that these domains are not restricted to a range (scope), and the discrete nature of the observation domain influences the motivation for interpolation schemes. For example, in sparse data recovery schemes, the observation domain can be defined in a discrete manner. Figure 3 provides a visualization of commonly used domain types for a selection of schemes which can employ *PEML* methods.

3. White box case: physics-based Bayesian filtering

Prior to overviews of the mentioned *PEML* classes and their adoption within the SHM and twinning context, we briefly recall a class of methods, which is situated near the *white-box* end of the spectrum in Figure 1, that is, Bayesian Filtering (BF). Perhaps one of the most typical examples of a hybrid approach to monitoring of dynamical systems is delivered in such BF estimators, which couple a system model (typically in state-space form), with sparse and noisy monitoring data. The employed state-space model can be either derived via a data-driven approach, for example, via the use of a system identification approach such as a Stochastic Subspace Identification (Peeters and De Roeck, 2001), or alternatively, it

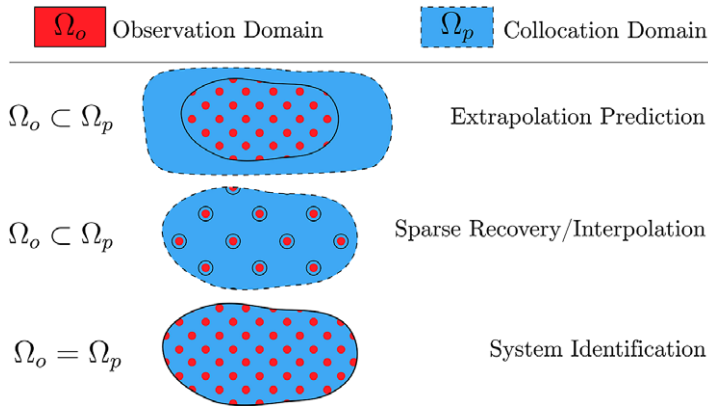


Figure 3. Visualization of domain definitions for schemes and motivations that can employ PEML. The blue areas represent the continuous collocation domain, and the red dots represent the coverage and sparsity of the discrete observation domain. The dashed and solid lines represent the scope of the collocation and observation domains, respectively.

may be inferred on the basis of a priori assumed numerical (e.g., finite element) model. We here refer to the former case, which we refer to as physics-based Bayesian Filtering. Such Bayesian filters can be used for estimation tasks of different complexity, including pure response (state) estimation, joint or dual state-parameter estimation (Chatzi and Smyth, 2009), input-state estimation (Eftekhar Azam et al., 2015; Maes et al., 2018; Sedehi et al., 2019; Vettori et al., 2023b), joint state-parameter-input identification (Dertimanis et al., 2019), or damage detection (Erazo et al., 2019). Bayesian filters draw their potency from their capacity to deal with uncertainties stemming from modeling errors, disturbances, lacking information on the structural system’s configuration, and noise corruption. However, they are limited by the requirement for a model structure, which should be representative of the system’s dynamics.

In the general case, the equation of motion of a multi-degree of freedom linear time-invariant (LTI) dynamic system can be formulated as:

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{D}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{S}_i\mathbf{f}(t), \tag{3}$$

where $\mathbf{u}(t) \in \mathbb{R}^{n_{dof}}$ is the vector of displacements, often linked to the Degrees of Freedom (DOFs) of a numerical system model, $\mathbf{M} \in \mathbb{R}^{n_{dof} \times n_{dof}}$, $\mathbf{D} \in \mathbb{R}^{n_{dof} \times n_{dof}}$, and $\mathbf{K} \in \mathbb{R}^{n_{dof} \times n_{dof}}$ denote the mass, damping, and stiffness matrices respectively; $\mathbf{f}(t) \in \mathbb{R}^{n_i}$ (with n_i representing the number of loads) is the input vector and $\mathbf{S}_i \in \mathbb{R}^{n_{dof} \times n_i}$ is a Boolean input shape matrix for load assignment. As an optional step, a Reduced Order Model (ROM) can be adopted, often derived via superposition of modal contributions $\mathbf{u}(t) \approx \Psi\mathbf{p}(t)$, where $\Psi \in \mathbb{R}^{n_{dof} \times n_r}$ is the reduction basis and $\mathbf{p} \in \mathbb{R}^{n_r}$ is the vector of the generalized coordinates of the system, with n_r denoting the reduced system dimension. This allows to rewrite equation (3) as:

$$\mathbf{M}_r\ddot{\mathbf{p}}(t) + \mathbf{D}_r\dot{\mathbf{p}}(t) + \mathbf{K}_r\mathbf{p}(t) = \mathbf{S}_r\mathbf{f}(t), \tag{4}$$

where the mass, damping, stiffness, and input shape matrices of the reduced system are obtained as $\mathbf{M}_r = \Psi^T\mathbf{M}\Psi$, $\mathbf{D}_r = \Psi^T\mathbf{D}\Psi$, $\mathbf{K}_r = \Psi^T\mathbf{K}\Psi$, and $\mathbf{S}_r = \Psi^T\mathbf{S}_i$.

Assuming the availability of response measurements, $\mathbf{x}_k \in \mathbb{R}^m$, at a finite set of m DOFs, such an LTI system can be eventually brought into a combined deterministic-stochastic state-space model, which forms the basis of application of Bayesian filtering schemes (Vettori et al., 2023a):

$$\begin{cases} \mathbf{z}_k = \mathbf{A}_d\mathbf{z}_{k-1} + \mathbf{B}_d\mathbf{f}_{k-1} + \mathbf{w}_{k-1} \\ \mathbf{x}_k = \mathbf{C}\mathbf{z}_k + \mathbf{G}\mathbf{f}_k + \mathbf{v}_k. \end{cases}, \tag{5}$$

where the state vector $\mathbf{z}_k = [\mathbf{p}_k^T \dot{\mathbf{p}}_k^T]^T \in \mathbb{R}^{2n_r}$ reflects a random variable following a Gaussian distribution with mean $\hat{\mathbf{z}}_k \in \mathbb{R}^{2n_r}$ and covariance matrix $\mathbf{P}_k \in \mathbb{R}^{2n_r \times 2n_r}$. Stationary zero-mean uncorrelated white noise sources \mathbf{w}_k and \mathbf{v}_k of respective covariance $\mathbf{Q}_k : \mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}_k)$ and $\mathbf{R}_k : \mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k)$ are introduced to account for model uncertainties and measurement noise. A common issue in BF schemes lies in calibrating the defining noise covariance parameters, which is often tackled via offline schemes (Odelson et al., 2006), or online variants, as those proposed recently for more involved inference tasks (Kontoroupi and Smyth, 2016; Yang et al., 2020; Vettori et al., 2023a).

Bayesian filters exploit this hybrid formulation to extract an improved posterior estimate of the complete response of the system \mathbf{z}_k , that is even in unmeasured DOFs, on the basis of a “predict” and “update” procedure. Variants of these filters are formed to operate on linear (Kalman Filter—KF) or nonlinear systems (Extended KF—EKF, Unscented KF—UKF, Particle Filter—PF, etc.) for diverse estimation tasks. Moreover, depending on the level of reduction achieved, BF estimators can feasibly operate in real, or near real-time. It becomes, however, obvious that these estimators are restricted by the rather strictly imposed model form.

In order to exemplify the functionality of such Bayesian filters for the purpose of system identification and state (response) prediction, we present the application of two nonlinear variants of the Kalman Filter on our Duffing oscillator working example. The system is simulated using the model parameters and inputs defined in Section 2. We further assume that the system is monitored via the use of a typical vibration sensor, namely an accelerometer, which delivers a noisy measurement of u . We further contaminate the simulated acceleration with zero mean Gaussian noise corresponding to 8.5% Root Mean Square (RMS) noise to signal ratio. For the purpose of this simulation, we assume accurate knowledge of the model form describing the dynamics, on the basis of engineering intuition. However, we assume that the model parameters are unknown, or rather uncertain. The UKF and PF are adopted in order to identify the unknown system parameters, namely the linear stiffness k , mass, m , and nonlinear stiffness k_3 . The parameter identification is achieved via augmenting the state vector to include the time-invariant parameters. A random walk assumption is made on the evolution of the parameters. The UKF employs a further augmentation of the state to include two dimensions for the process and measurement noise sources, resulting in this case in $2 \times 9 + 1 = 19$ Sigma points to simulate the system. It further initiates from an initial guess on the unknown parameters, set as: $k_0 = 1$ N/m, $c_0 = 0.5$ Ns/m, $k_{30} = 40$ N/m³, which is significantly off with respect to the true parameters. The PF typically employs a larger number of sample points in an effort to more appropriately approximate the posterior distribution of the state. We here employ 2000 sample points and initiate the parameter space in the interval $k \in \{520\}$ N/m, $c \in \{0.52\}$ Ns/m, $k_3 \in \{50160\}$ N/m³. In all cases, a zero mean Gaussian process noise of covariance $1e - 18$ (added to the velocity states) and a zero mean Gaussian measurement noise of covariance $1e - 18$ is assumed. Figure 4 demonstrates the results of the filter for the purpose of state estimation (left subplot) and parameter estimation (right subplot). The plotted result reveals a closer matching of the states for the UKF, while both filters sufficiently approximate the unknown parameters. More details on the implementation of these filters are found in Chatzi and Smyth (2009), Chatzi et al. (2010), and Kamariotis et al. (2023), while a Python library is made available in association with the following tutorial on Nonlinear Bayesian filtering (Tatsis et al., 2023).

Recent advances/applications of Bayesian filtering in structural mechanics include the following works. The problem of virtual sensing has been further explored in Tatsis et al. (2021, 2022) adopting a sub-structuring formulation, which allows to tackle problems, where only a portion of the domain is monitored. For clarity, sub-structuring involves dividing a complex domain into smaller, more manageable components, which are solved independently before integrated back into the full structure. Employing a lower level of reliance on the physics model form, by embedding physical concepts in the form of physics-domain knowledge, Tchemodanova et al. (2021) proposed a novel approach, where they combined a modal expansion with an augmented Kalman filter for output-only virtual sensing of vibration measurements. Greš et al. (2021) proposed a Kalman filter-based approach to perform subspace identification on output-only data, where in the input force is unmeasured. In this case, only the *periodic*

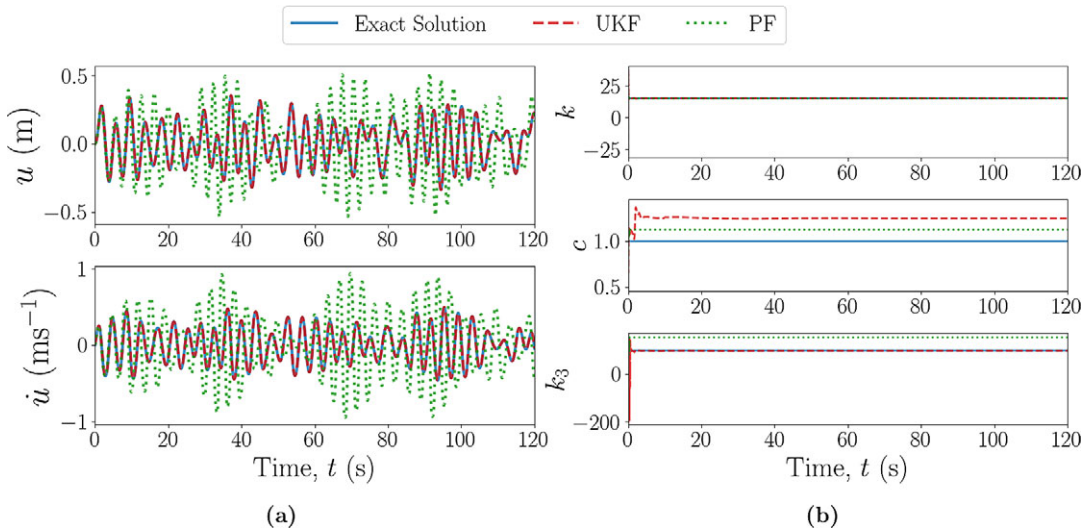


Figure 4. (a) State (response) estimation results for the nonlinear SDOF working example, assuming the availability of acceleration measurements and precise knowledge of the model form, albeit under the assumption of unknown model parameters. The performance is illustrated for use of the UKF and PF, contrasted against the reference simulation; (b) Parameter estimation convergence via use of the UKF and PF contrasted against the reference values for the nonlinear SDOF working example.

nature of the input force is known, and so this (unparameterized) information is also embedded within the model learning architecture. In comparison to filtering techniques with an assumed known force, this approach is less reliant on the physics model prescription, and so this approach has the advantage that it may be applied to a wide variety of similar problems/instances. The problem of unknown inputs has recently led to the adoption of Gaussian Process Latent Force Models (GPLFMs), which move beyond the typical assumption of a random walk model, that are meant to describe the evolution of the input depending on the problem at hand (Nayek et al., 2019; Rogers et al., 2020; Vettori et al., 2023b; Zou et al., 2023). Such an approach now moves toward a gray-like method (as discussed in the later sections), since Gaussian Processes, which are trained on sample data are required for data-driven inference and characterization of the unknown input model.

In relaxing the strictness of the imposed physics model, BF inference schemes can include model parameters in the inference task. Such an example is delivered in joint or dual state-parameter estimation methods (Dertimanis et al., 2019; Teymouri et al., 2023), which are further extended to state-input-parameter estimation schemes. In this context, Naets et al. (2015) couple reduced-order modeling with Extended Kalman Filters to achieve online state-input-parameter estimation, while Dertimanis et al. (2019) combine a dual and an Unscented Kalman filter, to this end; the former for estimating the unknown structural excitation, and the latter for the combined state-parameter estimation. Naturally, when the inference task targets multiple quantities, it is important to ensure sufficiency of the available observations, a task which can be achieved by checking appropriate observability identifiability, and invertibility criteria (Chatzis et al., 2015; Maes et al., 2019; Shi and Chatzis, 2022). Feng et al. (2020) proposed a “sparse Kalman filter,” using Bayesian logic, to effectively localize and reconstruct time-domain force signals on a fixed beam. As another example in the context of damage detection strategies, Nandakumar and Jacob (2021) presented a method for identifying cracks in a structure, from the state space model, using a combined Observer Kalman filter identification, and Eigen Realization Algorithm methods. Another approach to overcome to challenge of model-system discrepancy is to utilize ML approaches along with BF techniques to “bridge the gap,” but more will be discussed on this in Section 5, as these are no longer white-box models.

4. The black box case: deep learning models

Many modern ML methods are based on, or form extensions of, perhaps one of the most well-known methods, the neural network (NN). The NN can be used as a universal function approximator, where more complex models will generally require deeper and/or wider networks. When using multiple layers within the network, the method falls in the *deep-learning* (DL) class. For a regression problem, the aim of an NN is to determine an estimate of the mapping from the input \mathbf{x} , to the output \mathbf{y} . A fully connected, feed-forward NN is formed by N hidden layers, each with $n^{(N)}$ nodes. The nodes of each layer are connected to every node in the next layer and the values are passed through an activation function σ . For N hidden layers, the output of the neural network can be defined as,

$$\mathcal{N}_y(\mathbf{x}; \mathbf{W}, \mathbf{B}) := \sigma(\mathbf{w}^l x^{l-1} + \mathbf{b}^l), \quad \text{for } l = 2, \dots, N, \quad (6)$$

where $\mathbf{W} = \{\mathbf{w}^1, \dots, \mathbf{w}^N\}$ and $\mathbf{B} = \{\mathbf{b}^1, \dots, \mathbf{b}^N\}$ are the weights and biases of the network, respectively. The aim of the training stage is to then determine the network parameters $\Theta = \{\mathbf{W}, \mathbf{B}\}$, which is done by minimizing an objective function defined so that when the value vanishes, the solution is satisfied.

$$L_o = \langle \mathbf{y}^* - \mathcal{N}_y \rangle_{\Omega_o}, \quad \langle \cdot \rangle_{\Omega_x} = \frac{1}{N_k} \sum_{x \in \Omega_x} \|\cdot\|^2. \quad (7)$$

At the other end of the spectrum of a *white-box* (model-based) approach, where the system dynamics are transparent and therefore largely prescribed, thus lies a *black-box* approach, employing naive DL schemes to achieve stochastic representations of monitored systems. Linking to the BF structure described previously, Variational Autoencoders (VAE) have been extended with a temporal transition process on the latent space dynamics in order to infer dynamic models from sequential observation data (Bayer and Osendorfer, 2015). This approach offers greater flexibility than a scheme that relies on a prescribed physics-based model form, since VAEs are more apt to learning arbitrary nonlinear dynamics. The obvious shortcoming is that, typically, the inferred latent space need not be linked to coordinates of physical connotation. This renders such schemes more suitable for inferring dynamical features, and even condition these on operational variables (Mylonas et al., 2021), but largely unsuitable for reproducing system response in a virtual sensing context. Following such a scheme, Stochastic Recurrent Networks (STORN) (Bayer and Osendorfer, 2015) and Deep Markov Models (DMMs) (Krishnan et al., 2016), which are further referred to as Dynamic Variational Autoencoders (DVAEs), have been applied for inferring dynamics in a black box context with promising results in speech analysis, music synthesis, medical diagnosis and dynamics (Vlachas et al., 2022). In structural dynamics, in particular, previous work of the authoring team Simpson et al. (2021) argues that use of the AutoEncoder (AE) essentially leads in capturing a system's Nonlinear Normal Modes (NNMs), with a better approximation achieved when a VAE is employed (Simpson et al., 2023). It is reminded that, while potent in delivering compressed representations, these DL methods do not learn interpretable latent spaces.

In this paper, a rudimentary black-box method is demonstrated to provide a simple example of ML applied to the case scenario. Only one black-box approach is shown here to keep the focus overall to *PEML* techniques. Figure 5 shows the results of applying DMM to the working example. The 2σ uncertainty is also included, however, in this case, it is difficult to observe on the figure, as the uncertainty is small as a result of the low level of noise within the data. The data is generated for the time interval of 0 to 120 seconds with a sampling rate of 5 Hz and the displacement is assumed to be the only measurement. All the transition and observation models, as described by Krishnan et al. (2016), are modeled by black-box neural networks, specifically DMMs. While it is observed that the latent representation captures certain patterns of observed data, it lacks physical interpretability.

5. Light gray PEML schemes

When the prior physics knowledge of a system is relatively well-described, that is it captures most of the physics of the true system, it is possible to rely on this knowledge as a relatively strong bias, while further

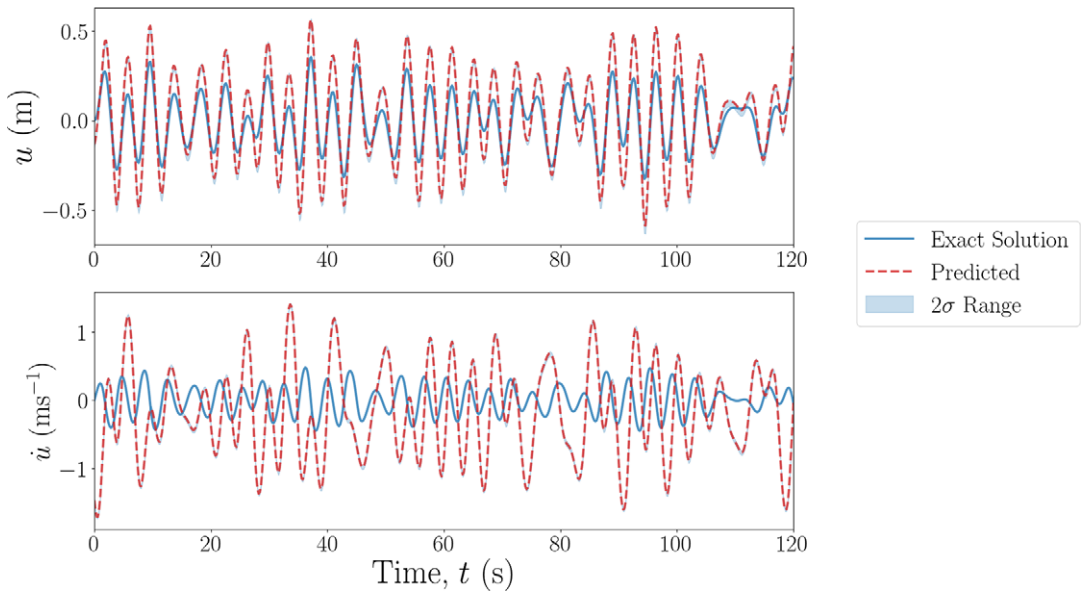


Figure 5. Predicted latent representations versus exact solutions of displacement (top) and velocity (bottom) using the DMM applied to the working example. Displacement is assumed to be the only measurement. The blue bounding boxes represent the estimated 2σ range.

exploiting learning schemes to capture any model mismatch. The term model mismatch or model-system discrepancy refers to the portion of the true system’s behavior (or response) which remains uncaught by the known physics. As a result of the larger degree of reliance on the physics-based model form, we here refer to this class of methods as “light-gray.” We will first discuss a survey of machine-learning-enhanced Bayesian filtering methods, which are still mostly driven by physics knowledge embedded in the BF technique. This is then followed by a section on Physics-Guided Neural Networks, which use the universal-approximation capabilities of deep-learning to determine a model of model-system discrepancy.

5.1. ML-enhanced Bayesian filtering

As previously stated, classical Bayesian filtering requires the model form to be known a priori, implying that the resulting accuracy will depend on how exhaustive this model is. To overcome the inaccuracies that result from model-system discrepancy, ML can be infused with BF techniques to improve inference potential. To this end, Tatsis et al. (2022) propose to fuse BF with a Covariance Matrix Adaptation scheme, to extract the unknown position and location of flaws in the inverse problem setting of crack identification, while simultaneously achieving virtual sensing. The latter is the outcome of a hierarchical BF approach powered by reduced order modeling.

Using a different approach, Revach et al. (2022) employ a neural network within a Kalman filter scheme to discover the full form of *partially* known and observed dynamics. By exploiting the nonlinear estimation capabilities of the NN, they managed to overcome the challenges of model constraint, that are common in filtering methods (Aucejo et al., 2019). Using a similar approach, but with a different motivation, Angeli et al. (2021) combined Kalman filtering with a deep-learning architecture to perform model-order reduction, by learning the mapping from the full-system coordinates, to a minimal coordinate latent space.

5.2. Physics-guided neural networks

In *physics-guided* machine learning (PGML), deep learning techniques are employed to capture the discrepancy between an explicitly defined model based on prior knowledge and the true system from

which data is attained. The goal is to fine-tune the overall model's parameters (i.e., the prior and ML model) in a way that the physical prior knowledge steers the training process toward the desired direction. By doing so, the model can be guided to learn latent quantities that align with the known physical principles of the system. This ensures that the resulting model is not only accurate in its predictions but also possesses physically interpretable latent representations. At this stage, we would like to remind the reader of the definitions of *physics model strictness* and *physics constraint* given in [Section 1](#). *PGML* approaches employ a relatively high reliance on the physics-based model form, in that the assumed physics is imposed in a strict form. However, in order to allow for simulation of model discrepancy, the level of physics constraint is relatively low, which means that the learner is not forced to narrowly follow this assumed prescribed form. The relaxation of constraints differs between *PGML* and physics-informed (*PIML*) approaches; *PGML* techniques often reduce constraints through the use of bias or residual modeling, whereas *PIML* schemes employ physics into the loss function as a *target* solution (i.e., the solutions are weakly imposed).

One of the key advantages of *physics-guided* machine learning is its ability to incorporate domain knowledge into the learning process. This is particularly beneficial in scenarios where data may be limited, noisy, or expensive to obtain. In comparison to *PEML* techniques that lay lower on our prescribed spectrum, *PGML* methods can offer an increased level of interpretability. By steering the model's learning process with physics-based insights, it can more effectively generalize to unseen data and maintain a coherent understanding of the underlying physical mechanisms.

The training process in *physics-guided* models involves two key components: (1) Incorporating Prior Knowledge: Prior knowledge on the physics of the system is integrated into the network architecture, or as part of the model; (2) Capturing Discrepancy: Deep learning models excel in learning from data, even when this contradicts prior knowledge. As a result, during training, such a model will gradually adapt and learn to account for discrepancies between the prior knowledge and the true dynamics of the system. This adaptability allows the model to converge toward a more accurate representation of the underlying physics.

Conceptually similar to estimating a residual modeler, Liu et al. (2021, 2022) proposed a probabilistic *physics-guided* framework termed a Physics-guided Deep Markov Model (PgDMM) for inferring the characteristics and latent structure of nonlinear dynamical systems from measurement data. It addresses the shortcoming of black-box deep generative models (such as the DMM) in terms of lacking physical interpretation and failing to recover a structured representation of the learned latent space. To overcome this, the framework combines physics-based models of the partially known physics with a DMM scheme, resulting in a hybrid modeling approach. The proposed framework leverages the expressive power of DL while imposing physics-driven restrictions on the latent space, through structured transition and emission functions, to enhance performance and generalize predictive capabilities. The authors demonstrate the benefits of this fusion through improved performance on simulation examples and experimental case studies of nonlinear systems.

Both residual modeling and *PgNNs*, in general, share a common objective of easing the training objective of neural networks. Residual modeling achieves this by easing the learning process through a general approximation, while *PgNNs* incorporate physical knowledge to provide reliable predictions even in data-limited situations. However, regarding specific implementations, there are differences between these two methods. Residual modeling operates without the need for specific domain knowledge, focusing instead on the residuals in a more general-purpose application. This makes it broadly applicable across various standard machine learning tasks, especially in the domain of computer vision. *PgNNs*, on the other hand, explicitly incorporate physical laws into the model, adding interpretability related to these physical models. *PgNNs* can also work in a general setting with any prior models to fit the residuals, but the key idea is to use physical prior model to obtain a physically interpretable latent representation from neural networks. This makes *PgNNs* particularly suited for problems where adherence to physical laws is paramount. Additionally, while residual networks rely heavily on data for learning, *PgNNs* can leverage physical models to make predictions even with limited data, demonstrating their utility in data-constrained environments.

To demonstrate how a physical prior model can impact the training of DL models, we apply the PgDMM to our working example, using the same data generation settings as explained in Section 4. In this case, instead of approaching the system with no prior information, we introduce a physical prior model into the DMM to guide the training process. This physical prior model is a linear model that excludes the cubic term in equation (1), which replicates a knowledge gap in the form of additional system complexities. The results are shown in Figure 6. It can be observed that the predictions for both displacements and velocities align well with the ground truth. The estimation uncertainty is slightly higher for velocities, which is expected since they are unobserved quantities. It is important to note that the system displays significant nonlinearity due to the presence of a cubic term with a large coefficient, causing the linear approximation to deviate noticeably from the true system dynamics. However, the learning-based model within the framework still captures this discrepancy and reconstructs the underlying dynamics through the guided training process.

A similar *physics-guided* RNN was proposed by Yu et al. (2020), which consists of two parts: physics-based layers and data-driven layers, where physics-based layers encode the underlying physics into the network and the residual block computes a residual value which reflects the consistency of the prediction results with the known physics and needs to be optimized toward zero.

Instead of modeling the residual of the prior model, a physics-guided Neural Network (PgNN), proposed by Karpatne et al. (2017), not only ingests the output of a physics-based model in the neural network framework, but also uses a novel physics-based learning objective to ensure the learning of physically consistent predictions, as based on domain knowledge. Similarly, the authors proposed a Physics-guided Recurrent Neural Network scheme (PGRNN, Jia et al., 2019) that contains two parallel recurrent structures—a standard RNN flow and an energy flow to be able to capture the variation of energy balance over time. While the standard RNN flow models the temporal dependencies that better fit observed data, the energy flow aims to regularize the temporal progression of the model in a physically consistent fashion. Furthermore, in another *PgNN* proposed by Robinson et al. (2022), the information from the known part of the system is injected into an intermediate layer of the neural network.

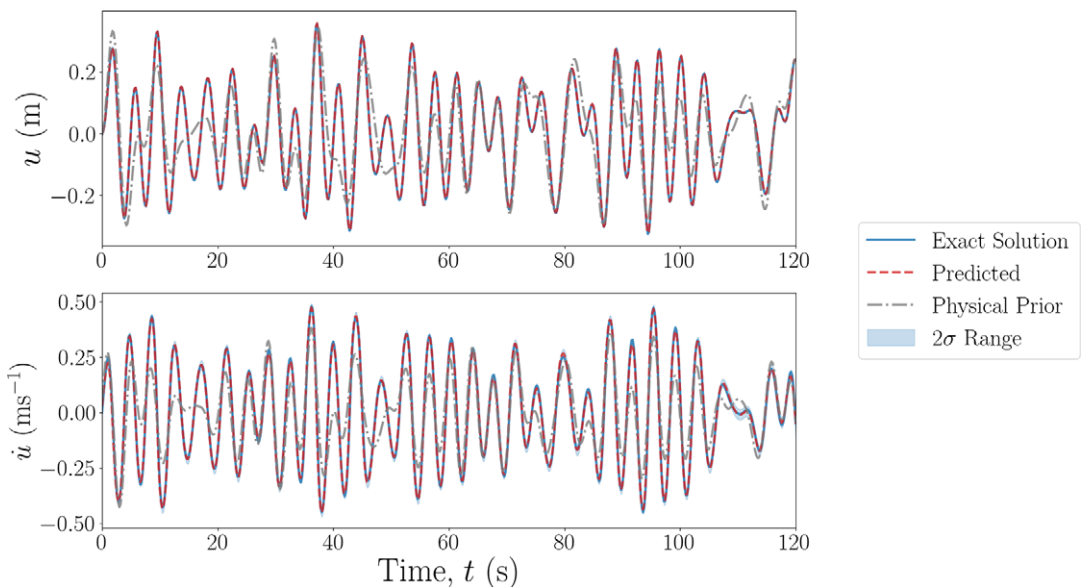


Figure 6. Predictions versus exact solutions of displacement (top) and velocity (bottom) using the PgDMM applied to the working example. Displacement is assumed to be the only measurement. The gray dash-dot line is the physical prior model and the blue bounding boxes represent the estimated 2σ range.

The physics-guided deep neural network (PGDNN), proposed by Huang et al. (2022), uses a cross-physics-data domain loss function to fuse features extracted from both the physical domain and data domain, which evaluates the discrepancy between the output of a FE model and the measured signals from the real structure. With the physical guidance of the FE model, the learned PGDNN model can be well generalized to identify test data of unknown damages. The authors also use the same idea in bridge damage identification under moving vehicle loads (Yin et al., 2023). Similarly, Zhang and Sun (2021) presented usage of the FE model as an implicit representation of scientific knowledge underlying the monitored structure and incorporates the output of FE model updating into the NN model setup and learning.

In Chen and Liu (2021), the physics knowledge is incorporated into the neural network by means of imposing appropriate constraints on weights, biases or both. Muralidhar et al. (2020) used physics-based prior model, physical intermediate variables, and *physics-guided* loss functions to learn physically interpretable quantities such as pressure field and velocity field.

6. Gray P_{EML} schemes

The motivations which drive the “middle ground” of *P_{EML}* will—naturally—vary depending on the nature of the information deficiency, and whether this is a physics knowledge gap or data scarcity. For example, the knowledge could be made up of a number of possible physical phenomena, or the system could be known but the parameters not. Or, with the opposite problem, it may be possible to capture data well in one domain, but be limited in the relative resolution of other domains (e.g., temporal vs. spatial). These are just a few of the many examples which motivate the use of “gray” *P_{EML}* techniques. In this section, two such approaches are surveyed and discussed; first dictionary methods are shown, which select a suitably sparse representation of the model via linear superposition from a dictionary of candidate functions. The second technique discussed is the physics-informed neural network, which weakly imposes conditions on the model *output* in order to steer the learner. This differs from the previously discussed physics-guided approaches, where the physics restrictions are strongly imposed by way of a proposed solution that instantiates an inductive bias to the learner. As will be discussed in detail below, the latter technique can be applied in a variety of ways, each embedding different prior knowledge and beliefs, allowing for flexibility in its application.

6.1. Dictionary methods

One of the biggest challenges faced in the practical application of structural mechanics in engineering, is the presence of irregular, unknown, or ill-defined nonlinearities in the system. Another challenge may occur from variation in the parameters which govern the prescribed model of the system, which could be from environmental changes, or from consequential changes such as damage. This motivates less reliance on the physics-based model form, to allow for freedom in physical-digital system discrepancy whilst satisfying known physics. One approach to reducing the reliance on the prescribed physics model form is by having the learner estimate the definition of the model, which may be the sole, or additional, objective of the learner. Dictionary methods are well-positioned as a solution for less strict physics embedding, where the model is determined from a *set* of possible model solutions, allowing freedom in a semi-discrete manner.

The problem of estimating the existence, type, or strength of the model-governing physics is described as that of *equation discovery*. This inverse problem can often be very computationally expensive, due to the large number of forward model calculations required to evaluate the current estimate of the parameters (Frangos et al., 2010). When determining the presence of governing equation components, often the identification is drawn from a *family* of estimated equations. For example, the matching-pursuit algorithm selects the most sparse representation of a signal from a dictionary of physics-based functions (Vincent and Bengio, 2002).

For dictionary-based approaches, the idea is to determine an estimate of the model output as some combination of bases or “atoms.” Often, these bases are formed as candidate functions of the input data,

and are compiled into a dictionary-matrix $\Theta(\mathbf{x})$. Then, linear algebra is used to represent the target signal from a sparse representation of this dictionary and a coefficient matrix Ξ .

$$\mathbf{y} = \Theta(\mathbf{x})\Xi. \quad (8)$$

And so, the aim of the learner is to determine a suitably sparse solution of Ξ , via some objective function. Typically, sparsity-promoting optimisation methods are used, such as LASSO (Tibshirani, 1996). The coefficient matrix can be constrained to be binary values, thus operating as a simple mask of the candidate function, or can be allowed to contain continuous values, and thus can simultaneously determine estimated system parameters which are used in the candidate functions. For the case of dynamic systems, a specific algorithm was developed by Brunton et al. (2016a) for sparse discovery of nonlinear systems. The team also showed how this could be used in control by including the control parameters in the dictionary definition (Brunton et al., 2016b). Kaiser et al. (2018) extended this nonlinear dictionary learning approach to improve control of a dynamic system where data is lacking. To do so, they extended the method to include the effects of actuation for better forward prediction.

In an example of dictionary-based learning, Flaschel et al. (2021) and Thakolkaran et al. (2022) showed a method for unsupervised learning of the constitutive laws governing an isotropic or anisotropic plate. The approach is not only unsupervised, needing only displacement and force data, it is directly inferrable in a physical manner. The authoring team then extended this work further to include a Bayesian estimation (Joshi et al., 2022), allowing for quantified uncertainty in the model of the constitutive laws.

Another practical example: Ren et al. (2023) used nonlinear dynamic identification to successfully predict the forward behavior of a 6DOF ship model, including coupling effects between the rigid body and water. In this work, the dictionary method is combined with a numerical method to predict the state of the system in a short time window ahead. This facet is often found when applying DM-based approaches to practical examples, as the method is intrinsically a model discovery approach, and so a solution step is required if model output prediction is required.

Data-driven approaches to equation discovery will often require an assumption of the physics models being exhaustive of the “true” solution; that is all the parameters being estimated will fully define the model, or the solution will lie within the proposed family of equations. One of the challenges faced with deterministic methods, such as LASSO (Tibshirani, 1996), is their sensitivity to hyperparameters (Brunton et al., 2016a); a potential manifestation of which is the estimation of a combination of two similar models, which is a less accurate estimate of the “true” solution than each of these models individually. Bayesian approaches can help to overcome this issue, by instead providing a stochastic estimate of the model, and enforcing sparsity (Park and Casella, 2008).

Fuentes et al. (2021) show a Bayesian approach for nonlinear dynamic system identification which simultaneously selects the model, and estimates the parameters of the model. Similarly, Nayek et al. (2021) identify types and strengths of nonlinearities by utilizing spike-and-slab priors in the identification scheme. As these priors are analytically intractable, this allowed them to be used along with a MCMC sampling procedure to generate posterior distributions over the parameters. Abdesslem et al. (2018) showed a method for approximate Bayesian computation of model selection and parameter estimation of dynamic structures, for cases where the likelihood is either intractable or cannot be approached in closed form.

6.2. Physics-informed neural networks

Raissi et al. (2019) showed that by exploiting automatic differentiation that is common in the practical implementation of neural networks, one can embed physics that is known in the form of ordinary/partial differential equations (ODEs/PDEs). Given a system of known ODEs/PDEs which define the physics, where the sum of these ODEs/PDEs should equal zero, an objective function is formed which can be estimated using automatic differentiation over the network. If one was to apply a PINN to estimate the state of the example in Figure 2, over the collocation domain Ω_c , using equation (2), the physics-informed loss function becomes,

$$L_p(\mathbf{t}; \Theta) = \langle \partial_t \mathcal{N}_z - \mathbf{A} \mathcal{N}_z - \mathbf{A}_m \mathcal{N}_{i^3} - \mathbf{B}f \rangle_{\Omega_c}, \tag{9}$$

where $\partial_t \mathcal{N}_z$ is the estimated first-order derivative of the state using automatic differentiation, and the system parameters are $\theta = \{m, c, k, k_3\}$. Boundary conditions are embedded into PINNs in one of two ways; the first is to embed them in a “soft” manner, where an additional loss term is included based on a defined boundary condition $\zeta(\mathbf{y})$ (Sun et al., 2020). Given the boundary domain $\partial\Omega \in \Omega_c$,

$$L_{bc} = \langle \zeta^* - \zeta(\mathcal{N}_y) \rangle_{\partial\Omega}. \tag{10}$$

The second case, so-called “hard” boundary conditions, involves directly masking the outputs of the network with the known boundary conditions,

$$\mathcal{N}_y = \zeta^* \Big|_{\partial\Omega}, \tag{11}$$

where often gradated masks are used to avoid asymptotic gradients in the optimization (Sun et al., 2020). In the case of estimating the state over a specified time window, the boundary condition becomes the initial condition (i.e., the state at $t = 0$).

$$\zeta = \mathbf{z}(0) = \{u(0), \dot{u}(0)\}^T, \quad L_{bc} = \langle \zeta - \mathcal{N}_z \rangle_{\Omega \in t=0}. \tag{12}$$

The total objective function of the PINN is then formed as the weighted sum of the observation, physics, and boundary condition losses,

$$L = \lambda_o L_o + \lambda_p L_p + \lambda_{bc} L_{bc}. \tag{13}$$

The flexibility of the PINN can then be highlighted by considering how the belief of the architecture is changed when selecting the corresponding objective weighting parameters λ_k . By considering the weighting parameters as 1 or 0, Table 1 shows how the selection of objective terms to include changes the application type of the PINN. The column titled Ω_c indicates the domain used for the PDE objective term, where the significant difference is when the observation domain is used, and therefore the PINN becomes akin to a system identification problem solution.

Figure 7 shows the framework of a generic PINN, highlighting where the framework, specifically the loss function formulation, can be broken down into the data-driven and physics-embedding components. It is also possible to include the system parameters θ as unknowns which are determined as part of the optimisation process, that is, $\Theta = \{\mathbf{W}, \mathbf{B}, \theta\}$. Doing so adds the capability as a system identification tool, as well as equation solution discovery. In system identification, either equation discovery can be performed, where the aim is to determine the definition of \mathcal{F} , or parameter estimation, where the aim is to determine the values of θ . The framework shown is for the training stage of the model, where the network parameters Θ , and optionally the physical parameters θ , are updated using an optimization algorithm such as LBFGS (Liu and Nocedal, 1989) or Adam (Kingma and Ba, 2014). Further prior physical knowledge can be added to PINNs by embedding more objective functions, such as initial conditions or continuity conditions, which can be formed directly from the output values of the network, or from the derivatives.

Table 1. Summary of PINN application types, and the physics-enhanced machine learning genre/category each would be grouped into

λ_{obs}	λ_{pde}	λ_{bc}	Ω_c	θ	PE-ML Subcategory
1	0	0	Ω_o	—	Purely data-driven
1	1	0/1	Ω_o	Unknown	System identification
1	1	1	Ω_c	Known	Physics-informed learner
0	1	1	Ω_c	Known	Forward-modeler

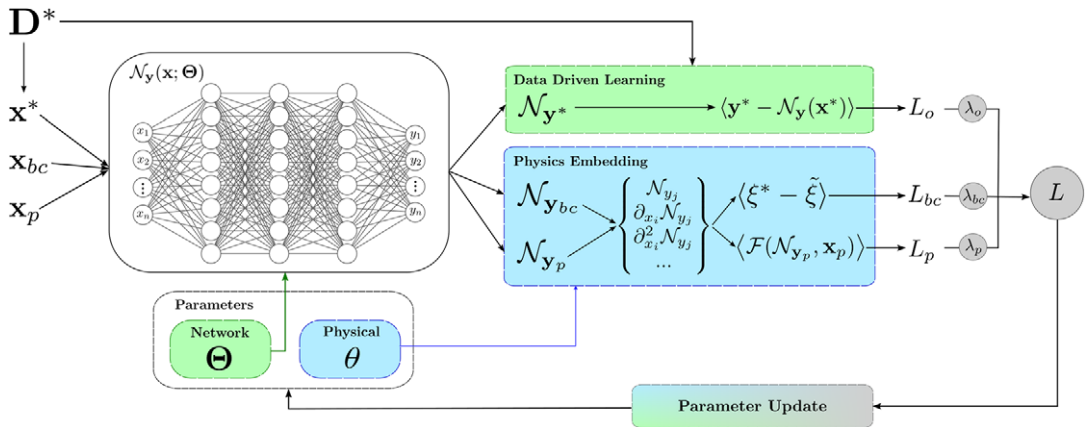


Figure 7. Framework of a general PINN, highlighting where the data-driven and physics-knowledge are embedded within the process.

To illustrate the different approaches to implementing a PINN, three paradigms of the method are discussed here, each to aiming to tackle a different objective.

1. *System Identification*; the aim is to determine the physics description of the system, either by equation discovery or by parameter estimation of the system parameters θ in equation (9) via estimating the system state.
2. *Enhanced Learning*; the aim is to enhance the learner to either better estimate the model given a more sparse set of data, or to improve learning efficiency.
3. *Forward Modeling*; where the PINN acts to generate a “simulated” model of the system, given the system equations and parameters, and the domain of interest.

These points will form the remaining parts of this subsection, where surveys will be shown and an example of each approach applied to the working example of the Duffing oscillator. The PINN is applied here as an “instance-modeler”; where the estimated model is only applicable in the case of the prescribed initial conditions and forcing signal. Therefore, new estimations made with this model are only applicable within the training domain. The generalisability and extrapolability presented with the PINN are in terms of extending beyond the domain of observations.

Physics-informed neural networks are gaining increased attention thanks to some of the advantages stated above, however, there exist a number of drawbacks which should be noted. Firstly, in many formulations of PINNs, they suffer from a lack of generalisability—which is a common motivation for PEML schemes—as they are restricted to the domain on which they were trained (Haghighat et al., 2021b). This domain may be extended beyond that of observations, however, computationally intensive training must still be performed before prediction. Furthermore, there is a lack of intuition or knowledge on the optimization task of PINNs; often, weighting of the losses is done via trial and improvement (Wang et al., 2021), and an under-defined, or ill-posed, physics prescription may result in many local minima (Nandi et al., 2021). A related challenge is in the computational effort of training, which can be a considerable task. This challenge has garnered fair criticism against the use of PINNs as opposed to other numerical-solving schemes (Grossmann et al., 2023).

6.2.1. PINNs for system identification

As discussed above, PINNs can be used as an approach to system identification, by applying a “soft” condition on the governing physical laws. The use of a variety of soft conditions, in the form of the loss functions, allows for discrepancies between the model and data, making them useful for noisy

observations. A good demonstration of *PINNs* for simultaneous state-system estimation is shown by Yuan et al. (2020) and Moradi et al. (2023), where in both examples, they model the displacement of a vibrating beam, with an accurate estimation of the governing equation parameters also. Zhang et al. (2020) use a *PINN* to solve an identification problem for nonhomogeneous materials, using elasticity imaging. An elegant approach here was done so by utilizing a multi-network architecture to include nonhomogeneous parameter fields, removing the potential issue caused by exploding dimensionality when including the spatially dependent material parameters. Sun et al. (2023) recently showed an approach to discovering the parameters of complex nonlinear dynamic systems using a Physics-informed Spline Learning approach. This approach employs the same exploitation as *PINNs*, however, the splines are used to allow for differentiation from a more sparse set of observed data, by interpolating the underlying dynamics.

Instead of determining the governing parameters of the PDEs, Wu and Xiu (2020) proposed a method to estimate the forward-in-time solution to a model. By recovering the evolution operator in the modal space, this reduces the problem from an infinite-dimension to a finite-dimension space. As opposed to value-based parameter identification, Ritto and Rochinha (2021) used measurement from a vibrating bar to update a ML-based classifier which directly infers the damage state of the structure. As the governing equations are often in the form of PDEs, an estimation of the derivatives is required in the learning process of many equation discovery approaches. However, Goyal and Benner (2022) proposed a numerical integration framework with dictionary learning, along with a “Runge–Kutta inspired” numerical procedure, overcoming the issues presented with derivative approximation from corrupted or sparse data.

For the working example, the *PINN* was applied to the Duffing oscillator, shown in Figure 2, and an equivalent linear system (i.e., $k_3 = 0$). Here, 256 data points were sub-sampled using a Sobol sampler and passed to the learner, and the physics-loss domain is set to the same as the observation domain (i.e., $\Omega_p = \Omega_o$), positing the framework akin to a system identification scheme. The observation loss (equation (7)) penalizes the output enough to result in an accurate estimation of the displacement, and satisfying the physics loss will drive the estimation of the physical parameters.

The results of the state estimation are shown in Figure 8, and the results of the parameter estimation are shown in Table 2. The state estimation results show accurate modeling, and the estimated values for the physical parameters also show a good level of accuracy. It is important to note here that accurate estimation of the state is not the primary objective, given that the domain of interest is well-covered by the observation data. A notable result is the increased accuracy when modeling only the linear system. This result can be explained by well-known machine learning intuition that with an increased dimensionality of estimation-space, an increased number of information is also required. Therefore, with the same *level* of information provided, the accuracy of the results for estimating more physical parameters will likely be more of a challenge for the learner.

6.2.2. *PINNs for enhanced learning*

Another utility of *PINNs* is domain enhancement potential, either by improving domain density from sparse observations, or by extending the domain beyond that of the observation domain. Practically for spatio-temporal models, increasing density is often only motivated in the spatial domain, as a result of the ease of improving time-domain sampling. However, domain-extension approaches are found to be practically motivated in both space and time.

Xu et al. (2022) used *PINNs* to accurately model the rigid-body dynamics of an unmanned surface vehicle as it voyages along a river. This is a nice example of a practical implementation of improving state prediction from sparse data, and a good example of how to formulate *PINNs* for relatively complex descriptions of dynamics. Chen and Liu (2021) use a *PINN* for estimating the fatigue $S - N$ curves, where even on seemingly sufficient data, the uninformed ANN fails to accurately predict. A particular note of this work, was the inclusion of a probabilistic framework, allowing both freedom in the model construction, as a result of stochastic considerations, and a quantified estimate of the uncertainty. By utilizing a finite-element model, in which the parameters were updated using a *PINN*, Zhang and Sun (2021) developed a NN-based method for detecting damage.

Table 2. Results of system estimation for the SDOF oscillator for both the nonlinear and linear case

System Case	c		k		k_3	
	Predicted	Error	Predicted	Error	Predicted	Error
Nonlinear	1.002	0.26%	15.02	0.12%	99.22	0.78%
Linear	1.002	0.213%	15.00	0.011%	-	-

Note. For all parameters, the estimated value and the percentage error are shown.

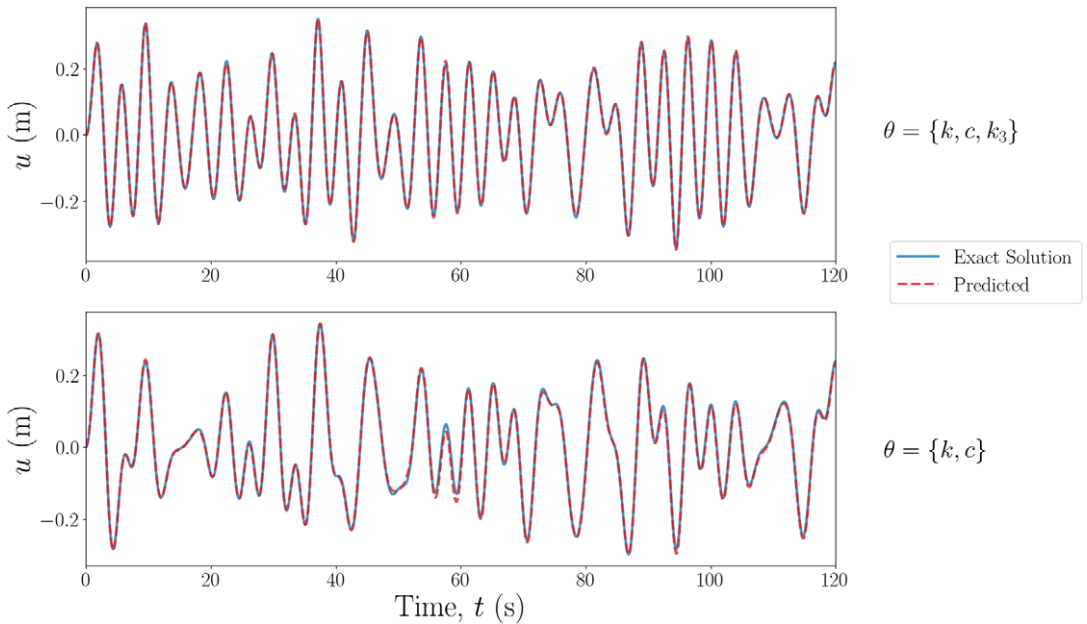


Figure 8. Predicted versus exact solution of simultaneous system-state estimation approach to solving the working example for the nonlinear case (top) and linear case ($k_3 = 0$) (bottom).

By utilizing an energy-based formulation of the loss function, Zhuang et al. (2021) modeled bending, vibration, and buckling of a Kirchoff plate. As well as the informed loss function, the authors used a non-standard activation function to better emulate the underlying operations which govern the physics. Another example of an energy-based loss function approach is shown by Goswami et al. (2020), who combined a *PINN* with transfer learning to model the phase field of fracture in a material. They showed how a well-trained model could drastically reduce the computational requirements of this problem.

The deformation of elastic plates with *PINNs* was shown by Li et al. (2021), where they made a comparison between purely data-driven, PDE-based, and energy-based physics informing, finding each to have different advantages. The PDE-based approach was less dependent on sampling size and resolution, whereas the energy-based approach had less hyperparameter, and therefore was more efficient and easier to train. This information may be useful for the future development of *PINNs*, which suffer from a lack of understanding of the hyperparameter space pathology, to allow for robust optimization strategies, as discussed by Wang et al. (2021).

Yucesan and Viana (2020) present a methodology for predicting the damage level in a wind turbine blade, in the form of grease degradation, using *PINNs*. In their work, the advantage of *PINNs* is that instead of aiming to directly model the value of grease degradation, they aim to predict the *increment* of grease degradation based on the current value and a number of other measurable quantities. This would be

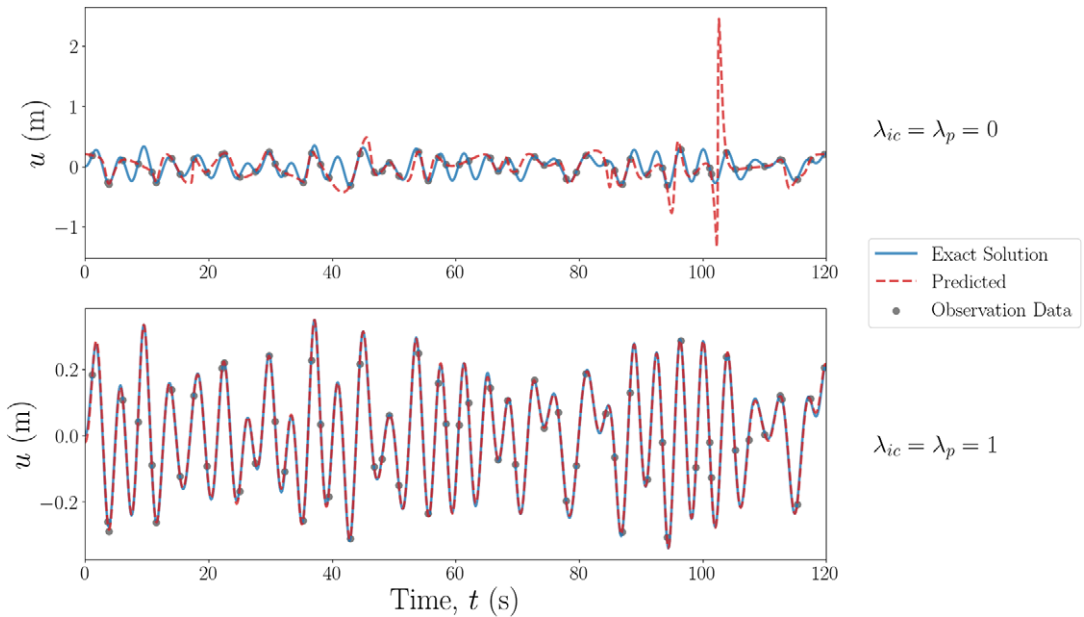


Figure 9. Predicted versus exact solution of state estimation approach applied to a subsample of the working example with no physics embedded (top) and physics-informed embedding (bottom).

a difficult task to perform in a black-box manner, as the values of the increment are not the observed values, and so the physics embedding helps to overcome this.

Now, we will show the *PINN* applied as an enhanced learner to the working example, where the physical parameters are known a priori. The advantages of the *PINN* in this aspect are more efficient learning, and for improved extrapolation of the data from sparse observations. To demonstrate, only every sixteenth sample is instead fed to the learner, emulating a sampling frequency of 0.5328 Hz. Then, the *PINN* is applied as a “black-box” architecture (i.e., $\lambda_{ic} = \lambda_p = 0$), and as a physics-informed modeler, where all loss weight parameters are included. The results of the two methods applied are shown in Figure 9, where it is clear to see the strength of embedding physics, demonstrated in the context of sparse data.

6.2.3. PINNs for forward modeling

When applying *PINNs* for forward modeling of systems and structures, no prior observations are given, that is, $D^* = \emptyset$, however, it is necessary to provide sufficient boundary and initial conditions, and physics constraints which describe a complete model. An advantage of *PINNs* for forward modeling is their simple implementation; embedding boundary conditions, complex geometries, or new governing equations is relatively straightforward. The training time of *PINNs* as forward modelers, in relation to traditional finite-element methods, is often greater, leading to statements implying their impracticality (Rezaei et al., 2022). In the context of forward modeling of a known model, and at pre-set collocation points, this statement of impracticality is well-founded. However, *PINNs* may also provide a convenient solution for problems such as; in-time control/prediction, where a *PINN* could predict an output rapidly, as the computational effort is done a priori during training, or for determining a better generalization over high-dimensional modeling space, where the computational cost of FEM methods can rapidly increase with dimensionality.

PINNs for forward modeling have gained a lot of traction in micro-scale problems; Haghghat et al. (2021b) and Henkes et al. (2022) use *PINNs* to model the instance of displacements and stresses in a unit cell. In the latter, they showed the capability of the approach to model nonlinear stresses by including a sharp phase transition within the material. Haghghat and the team also showed how the method could

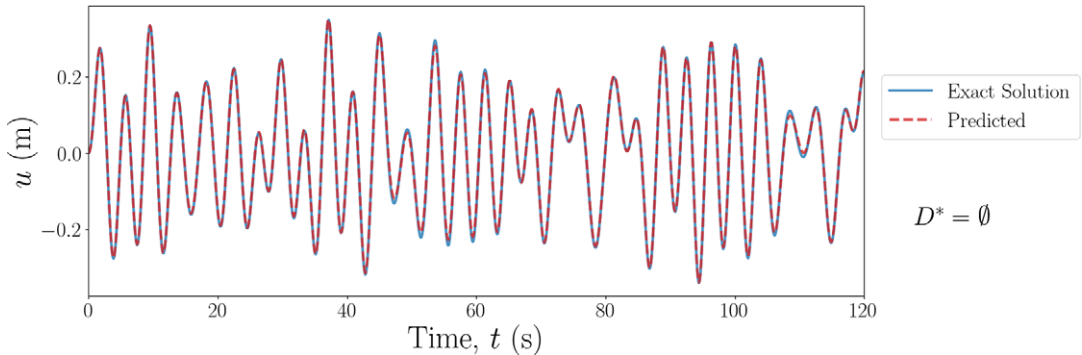


Figure 10. Exact solution versus PINN-based forward modeling solutions of the SDOF Duffing oscillator example, where no observations of the state are given to the learner.

accurately model structural vibration (Haghighat et al., 2021a), giving only initial and boundary conditions. Abueidda et al. (2021) showed *PINNs* for modeling various solid mechanics effects; elasticity, hyperelasticity, and plasticity, where the method performed well on all types of materials. In the work by Zheng et al. (2022), fracture mechanics were modeled to a decent accuracy with only principle physics.

The above examples all follow a fairly straightforward path to the model formulation, by employing the variables that are inherent to the governing equations as the variables of the ML model. Going beyond this, Huang et al. (2020) used a *proper orthogonal decomposition* (POD) neural network to model plasticity in a unit cell. The advantage of the POD approach is to decouple the multi-dimensional stress, allowing the use of individual NNs for each stress variable, reducing computation time, and increasing learning efficiency. Straying away from the common approach of using PDEs to form the physics-based loss, Abueidda et al. (2022) formulated an energy-based loss term to successfully forward model hyperelasticity and viscoelasticity in a given material.

As a final demonstrator of *PINN* utility, we apply it as a forward modeler, with no observed data offered to the learner, that is, $\Omega_o = \emptyset$. However, it is important to note that there is still *training data* reflecting time domain information. For this implementation, initial conditions, in the form of Dirichlet and Neumann boundary conditions from equation (12), and the forcing signal are provided. In Figure 7, therefore, only the physics embedding portion of the framework is included. The results of this forward modeling approach are shown in Figure 10, where it can be seen the solution matches well with the exact solution.

7. Dark gray PEMPL schemes

In this final section of the *physics-enhanced* discussion, the “darkest” genres of *PEMPL* techniques (that are included in this paper) are discussed. These techniques have a lower reliance on the physics-based model form, as the prescribed model has a lower level of strictness.

7.1. Constrained GPs

One example of *physics-encoded* learners is constrained Gaussian processes (GPs), which, depending on kernel design, can be viewed as embedding the general shape of the function, as GPs are a problem of discovery over the *function-space*, as opposed to the *weight-space* (O’Hagan, 1978; Williams and Rasmussen, 2006). Conceptually, this process can be thought of as estimating a distribution over all the possible functions that could explain the data, as opposed to one “best fit” model. The aim is to estimate a nonlinear regression model given a set of observed output data \mathbf{y} , and observed input data \mathbf{x} ,

$$\mathbf{y} = f(\mathbf{x}) + \varepsilon, \quad \varepsilon : \mathcal{N}(0, \sigma_n^2 \mathbb{I}), \quad (14)$$

where ε is a zero-mean Gaussian white-noise process with variance σ_n^2 . A GP is fully defined by its mean and covariance functions,

$$f(\mathbf{x}) : \mathcal{GP}(m(\mathbf{x}), K(\mathbf{x}, \mathbf{x}')). \quad (15)$$

The mean function $m(\mathbf{x})$ can be any parametric mapping of \mathbf{x} , and the covariance function expresses the similarity between two input vectors \mathbf{x} and \mathbf{x}' . The primary influence of the user, when implementing a GP, is in the choice of the covariance kernel, which is calculated as any other kernel; linear pair-wise distances between points to form a covariance matrix. There are a number of popular kernels that are used as standard, each of which embed a different belief as to which *family of functions* the model solution is drawn from. There are two primary methods of embedding physical knowledge in GPs; the first of which is to include an initial estimate of the model into the mean function, and so the problem can then be envisioned as determining the solution of the remaining, unknown physics.

The second approach is to design, or select, the covariance kernels to constrain the shape of the function estimate, and often combinations of kernels can provide varying levels of physical knowledge embedding (Cross and Rogers, 2021). In this case, it is possible to relax the reliance on the physics model form, by for example simply dictating the design of the kernel for the domain on which the physics is expected to operate. For example, Padonou and Roustant (2016) showed a kernel for predicting on circular domains. More strictly prescribed model forms can be applied by designing kernels to include physical knowledge in the form of partial differential equations, or boundary conditions (Solin and Särkkä, 2020).

An example of embedding prior knowledge via the mean function is shown by Zhang et al. (2021), where they showed an improved GP model for the deflection of the Tamar bridge (Cross et al., 2013) under time-varying environmental conditions, by including an expected linear deflection of the cable due to temperature. Data from the Tamar bridge represents a relatively complex problem; a large-scale complicated structure under varying (and non-exhaustive) environmental conditions. However, even by applying the simplest method of embedding physics into a GP, the modeling was improved. Petersen et al. (2022) also applied a novel physics-informed GP method to a bridge problem, but with the aim of estimating wind load from acceleration data. In their work, they developed a novel infusion of GP latent-force model (GP-LFM) with a Kalman filter-based approach. The inclusion of the GP-LFM allowed for characterization of the *evolution* of the wind-load, and this is enriched with prior physical knowledge in the form of stochastic information on wind-loads taken from wind-tunnel tests. This work provides an excellent demonstration of how physics information can be embedded to allow the transfer of information from scaled structures.

Haywood-Alexander et al. (2021) used constrained GPs to model the physical characteristics of guided-waves in a complex material. Guided waves in complex materials are famously difficult to model due to their relatively short wavelength in comparison to the material structure. By designing a variety of kernels, they demonstrated the performance of the GP modeling with varying levels of physics information embedded. Notably, they showed that by considering the space in which the physics operates, one can already improve the modeling capabilities, even before any physics equations are embedded. Continuing on the topic of elastic waves, Jones et al. (2023) applied PI-GPs to localize acoustic source emission in a complex domain. They developed constrained GPs further by embedding boundary conditions and the spatial domain of the problem. This approach has potential for use in modeling on structures with relatively high geometrical complexity, such as those with lots of joints, or with layered materials.

As stated above, the GP is unique in that it operates in the *function* space, thus, prior knowledge embedded is on that of the shape of the function. This characteristic was utilized by Dardeno et al. (2021), who used weak-form dynamics equations as a mean function within a novel overlapping mixture of Gaussian processes (OMGP) method. By constraining the expected shape of the functions, this allows the learner to separate out unsorted data of dynamic structures from within a population.

To demonstrate the constrained GP, here we apply this to the working example in Figure 2. In this context, the GP estimates a nonlinear operator where the input is time (i.e., $\mathbf{x} = \mathbf{t}$). The first kernel we will

select is the scaled squared-exponential kernel, where l is a lengthscale hyperparameter, and α is the scaling hyperparameter,

$$K_{SE}(\mathbf{t}, \mathbf{t}^*) = \exp\left(-\frac{1}{2l^2}(\mathbf{t}-\mathbf{t}^*)^T(\mathbf{t}-\mathbf{t}^*)\right), \tag{16}$$

which embeds only a belief that the function is smooth. Then, following the work of Cross and Rogers (2021), if we assume a Gaussian white noise force input, an additional physics-derived kernel can be included,

$$K_{SDOF}(\mathbf{t}, \mathbf{t}^*) = \frac{\sigma_f^2}{4m^2\zeta\omega_n^3} e^{-\zeta\omega_n|\tau|} \left(\cos(\omega_d\tau) + \frac{\zeta\omega_n}{\omega_d} \sin(\omega_d|\tau|) \right), \quad \tau = \mathbf{t} - \mathbf{t}^*, \tag{17}$$

where σ_f is an additional hyperparameter, representing the square root of the variance of the forcing. To demonstrate the constrained GP approach, the data from Figure 2 is applied in a similar fashion to the informed-learner approach used with the PINN in Figure 9. Here, only every twelfth sample is fed to the learner, and is first applied with the uninformed squared-exponential kernel in equation (16), and then with the constrained SDOF kernel in equation (17). The results of the estimated solutions using each of these kernels are shown in Figure 11, along with the estimated 95% (2σ) confidence range. Note, the value of σ here is taken from the estimated covariance matrix, and is not the hyperparameter σ_f . It is clear to see the improved estimation of the displacement of the system, but on top of this, the estimated uncertainty is also reduced. An interesting observations from this is that when using a linear SDOF kernel to model a nonlinear system, this still results in improved modeling from sparse data.

Constrained Gaussian processes are often used to improve the forecasting of temporal data, which falls under the umbrella of domain extension schemes. In the context of the example prescribed here, for accurate forecasting one would need to determine an analytical solution for the covariance of a forced nonlinear system, as with the prescribed kernel, prediction beyond the scope of the data results in simply a free-vibration system.

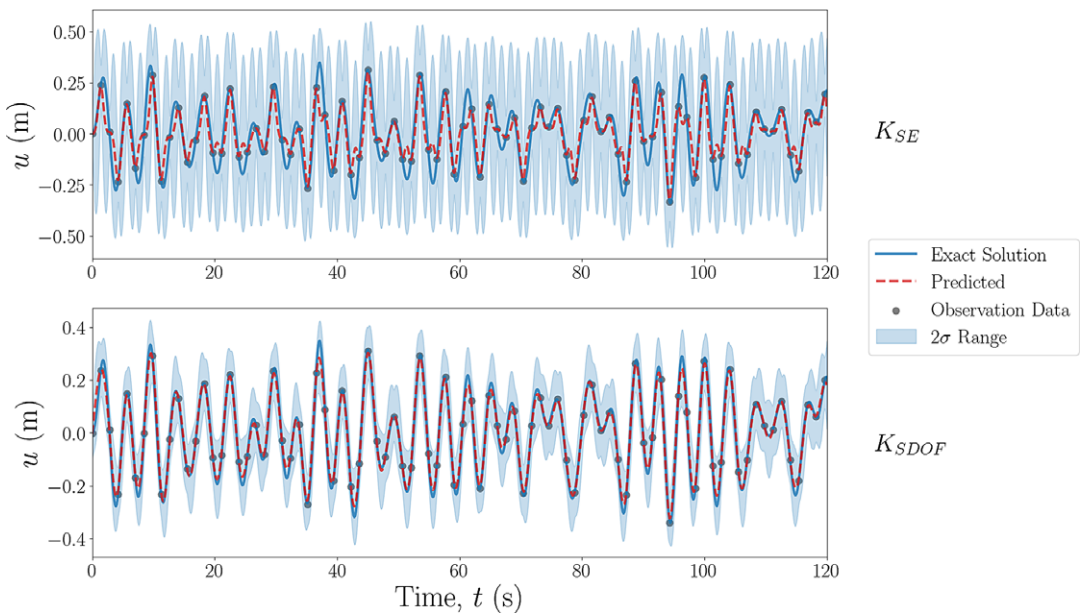


Figure 11. Predicted versus exact solutions of displacement estimation using a GP applied to a subsample of the working example, with (top) no physics embedded and (bottom) constrained GP. The blue bounding boxes represent the estimated 2σ range.

7.2. *Physics-encoded neural networks*

So far we have only explored how the automatic differentiation mechanisms can help embed physics into a neural networks. This process can be applied blindly to any type of neural network architecture. A different approach is to modify the architecture of the neural network itself such that its hidden states conform to a domain closer to that of the physical problem of study. By imposing specific geometric constraints, it is possible to bias neural networks toward such domains. Adding additional biases to the model reduces the variance during the training phase, which leads to a faster convergence. However, such biases must be carefully imposed such that they replicate the physics of the system of study, else they add a constant bias error into the model.

The biases that one can introduce in a machine learning model are in most cases symmetries that are imposed on the nonlinear mappings of the model (Bronstein et al., 2021). The rationale for imposing symmetries stems from Noether's theorem: every continuous symmetry in a physical system corresponds to a conserved quantity. Therefore, symmetries in a model's architecture should be able to encode the physical properties of the system that are conserved. From henceforth, we will refer to models as "*physics-encoded*" whenever their architecture is biased to specifically reproduce the symmetries arising from the properties of the system of study. Such symmetries can be tailored depending on the prior knowledge available on the system.

7.2.1. *Neural ODE*

Neural ordinary differential equations (Chen et al., 2018) are a framework that unifies neural networks and ordinary differential equations to model dynamical systems. Unlike traditional neural networks that operate on discrete time steps, Neural ODEs model the evolution of a system continuously over time. They leverage the powerful tools of ODE theory to learn and infer hidden states, trajectories, and dynamics from observed data. At the core of Neural ODEs is the use of the adjoint sensitivity method, which enables efficient gradient computation. This technique allows gradients to be backpropagated through the ODE solver, enabling end-to-end training of the model using standard gradient-based optimization algorithms.

Neural ODEs offer several advantages over traditional neural networks. Firstly, they provide a flexible and expressive modeling framework that can capture complex temporal dependencies and nonlinear dynamics. Secondly, they inherently handle irregularly sampled or sparse data since the ODE solver can handle time interpolation. This is particularly valuable when dealing with real-world physical systems where data may be scarce or unevenly sampled. Lastly, Neural ODEs can exploit known physical laws or priors by incorporating them into the ODE function, thereby enhancing the interpretability and generalization of the model.

Physics-encoded machine learning using Neural ODEs has found applications in various domains, such as fluid dynamics, particle physics, astronomy, and material science. By incorporating physical knowledge into the model architecture, Neural ODEs can leverage the underlying physics to improve predictions, simulate systems, and discover new phenomena. Using *PeNNs* in an equation-discovery context, Lai et al. (2021) utilize physics-informed Neural ODEs for a structural identification problem, where varying levels of prior constraints are embedded in the learner. They showcased a framework that allows for an inferrable model, by adopting a sparse identification of nonlinear dynamical systems. Further works by Lai et al. (2022) show that such a scheme can be integrated into generative models such as VAEs. This framework leverages physics-informed Neural ODEs via embedding eigenmodes derived from the linearized portion of a physics-based model to capture spatial relationships between DOFs. This approach is notably applied to virtual sensing, that is, the recovery of generalized response quantities in unmeasured DOFs from spatially sparse data.

7.2.2. *Hamiltonian neural networks*

An alternative method for embedding physics into the architecture of the network is to constraint them according the Hamiltonian formalism. The intuition behind this method originates from Noether's first

theorem which states that every differentiable symmetry of the action of a physical system with conservative forces has a corresponding conservation law. In layman terms, this means that the symmetries that are observed within the dynamics of the system are the result of a conservation of the properties of said system.

With the success of PINNs, more and more theoretical research on the integration of physical principles has been taking place. Namely, how do different formulations of a system’s equations can be used as prior knowledge to bias our model. The most common way in mechanics is to represent a system as its state-space. An alternative formulation can be done from the point of view of the energy of the system through the Hamiltonian formulation. For our state-space \mathbf{z} in the case of a MDOF formed by the pair (\mathbf{q}, \mathbf{p}) , the Hamiltonian is formulated as:

$$H(\mathbf{q}, \mathbf{p}) = \frac{1}{2} \mathbf{p}^T \mathbf{M}^{-1}(\mathbf{q}) \mathbf{p} + V(\mathbf{q}). \tag{18}$$

With $\mathbf{M}^{-1}(\mathbf{q}) \succeq 0$. The Hamiltonian is considered to be separable when $H(\mathbf{q}, \mathbf{p}) = T(\mathbf{p}) + V(\mathbf{q})$. When looking at the variations of the Hamiltonian over time, one notices that $\dot{\mathbf{q}} = \frac{\partial H}{\partial \mathbf{p}}$ and $\dot{\mathbf{p}} = -\frac{\partial H}{\partial \mathbf{q}}$. This leads to the Hamiltonian being time-invariant since:

$$\dot{H} = \left(\frac{\partial H}{\partial \mathbf{q}} \right)^T \dot{\mathbf{q}} + \left(\frac{\partial H}{\partial \mathbf{p}} \right)^T \dot{\mathbf{p}} = 0. \tag{19}$$

Greydanus et al. (2019) are the first to have introduced the Hamiltonian formalism to neural networks to bias the model for physical data. The method is closer to a Physics-informed than to a Physics-encoded model since the formalism is added through the loss function. The Hamiltonian loss is given by (we remind the reader that Θ are the parameters of the model):

$$\mathcal{L}_{\text{HNN}} = \left\| \frac{\partial H_{\Theta}}{\partial \mathbf{p}} - \frac{\partial \mathbf{q}}{\partial t} \right\|_2 + \left\| \frac{\partial H_{\Theta}}{\partial \mathbf{q}} - \frac{\partial \mathbf{p}}{\partial t} \right\|_2. \tag{20}$$

In this case, is the neural network is learn the gradients of the system. Such a formalism prevents to the neural network prediction to stray away from the true state of the system by grounding it in the physical domain.

This property can also be found in symplectic integrators, that is, integrators derived from the Hamiltonian formalism. For a given initial value problem, a discrete integration of the system can be performed with explicit integrators such as Euler integration. Such integration relies on the local Taylor expansion of the flow of the system. For an integration of the n th order, an integration error of the $n + 1$ -th order accumulates at every time step, leading to a drift from the true dynamics of the system. This drift can be avoid by opting for symplectic gradients instead. The first-order symplectic gradients can be derived as follows:

$$\dot{\mathbf{q}} = \frac{\partial H}{\partial \mathbf{p}} \Rightarrow \dot{\mathbf{q}}_t = \frac{\mathbf{q}_{k+1} - \mathbf{q}_k}{h} \Rightarrow \mathbf{q}_{k+1} = \mathbf{q}_k + h \frac{\partial H}{\partial \mathbf{p}_k}. \tag{21}$$

$$\dot{\mathbf{p}} = -\frac{\partial H}{\partial \mathbf{q}} \Rightarrow \dot{\mathbf{p}}_t = \frac{\mathbf{p}_{k+1} - \mathbf{p}_k}{h} \Rightarrow \mathbf{p}_{k+1} = \mathbf{p}_k - h \frac{\partial H}{\partial \mathbf{q}_k}. \tag{22}$$

For t the continuous time, k the discrete time, and h the discretisation. For the coefficients of symplectic integrators of the n th order (equivalent to their explicit counterparts), we refer to the formula derived by Yoshida (1990) (c_i and d_i given in the paper):

$$e^{h(T(\mathbf{p})+V(\mathbf{q}))} = \prod_{i=1}^n e^{c_i h T(\mathbf{p})} e^{d_i h V(\mathbf{q})} + \mathcal{O}(h^{n+1}). \tag{23}$$

The symplectic integration principle has been extended to a plethora of *physics-encoded* models. Chen et al. (2020) modified the original HNN by Greydanus by replacing the ANN with an RNN and by

updating the gradients in a symplectic manner. Saemundsson et al. (2020) train a Neural ODE with a split latent space with symplectic integrators. Sanchez-Gonzalez et al. (2019) estimate the Hamiltonian of the system, then derive the gradients from its estimate to update the model. David and Méhats (2021) employ the same principle but replace the ANN with a Graph Neural Network (GNN). One of the issues with the Hamiltonian approach is that it assumes constant energy within the system, something that is usually not the case for most real-world applications. Many methods attempt to resolve this issue by incorporating dissipation into their formulation. Sosanya and Greydanus (2022), an adaptation of the original HNN that splits the gradients into their dissipative and non-dissipative components. Desai et al. (2021) adopt the port-Hamiltonian formalism to adopt HNNs, making them apt to learn the dynamics of control systems.

The first work to extend this notion to DL is that of Greydanus et al. (2019), which enforces a symmetric gradient on a neural network trained to predict the dynamics of a conservative system. Saemundsson et al. (2020) showed that the Hamiltonian formalism could be combined with the previously mentioned Neural ODEs, yielding the so-called Symplectic (state-space area preservation) Neural ODEs. Zhong et al. (2020) also used neural ODEs to learn the physics in an inferrable manner, applying Hamiltonian dynamics. Particularly, they parameterized the model in order to enforce Hamiltonian mechanics, even when only velocity data can be accessed as opposed to momentum. Bacsa et al. (2023) propose a method extending this reasoning to stochastic learning, where a symplectic encoder learns an energy-preserving latent representation of the system, and opens up new considerations for physics-embedded NN architectures.

We extend our SDOF oscillator example to other tasks to demonstrate the use of Neural ODEs. Neural ODEs are trainable forward models in that a neural network is used to approximate the flow of the system of study. In this context, the neural ODE estimates the latent space \mathbf{z}_t given the initial value problem starting at (\mathbf{z}_0, t) . The Neural ODE flow estimation is done using the ResNet (He et al., 2015) such that the integration is accumulated on top of the residuals of the neural network. The neural network is optimized using the *adjoint sensitivity method* (Pontryagin et al., 1962). The results are seen in Figure 12.

We can extend this method within the DMM framework, using the symplectic Neural ODEs, as per Bacsa et al. (2023). We change the problem accordingly: given that symplectic networks are made to deal with limit cycles, the forcing is switched from white excitation to a multisine excitation. The results are seen in Figure 13.

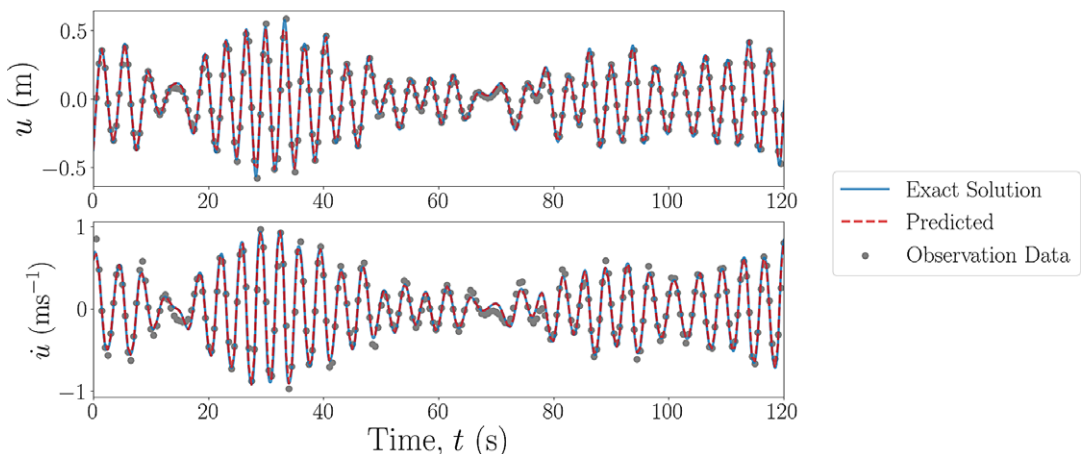


Figure 12. Predicted versus exact solutions of state-space estimation of the Neural-ODE $k + 1$ predictor.

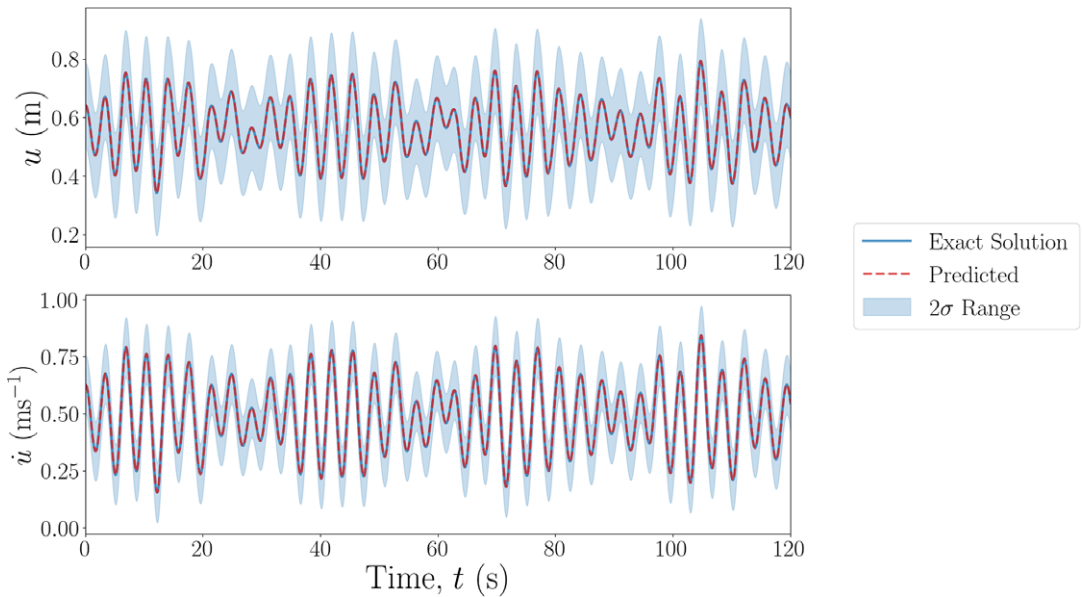


Figure 13. Predicted versus exact solutions of state-space estimation of the Symplectic Neural-ODE encoded DMM $k + 1$ predictor with uncertainty.

8. Discussion

In Section 1, we discussed variants situated across the spectrum of *PEML*, and examined the characterization of such methods based on their reliance on the prescription of the physics-based model form (and physics constraints) embedded within the learner, and the amount of data used. The selection of an appropriate scheme is driven by the motivation or, in other words, the nature of the downstream task, the level and type of prior scientific/physical knowledge, and the amount of data available. When the true system is unknown, or much too complex to define an adequate model, purely data-driven (black-box) models are used. As these models are extremely non-generalizable and are limited to only the scenario for which the data has been collected, they can only reasonably be applied in a sufficiently similar scenario. Furthermore, such types of models require training on large amounts of data for reasonable accuracy. The advantage, however, of such models is their extreme levels of flexibility; they embed zero prior belief of the true system, and are often described as universal approximators.

When the true system is relatively simple, and can be adequately modeled with only prior scientific knowledge, white-box models can be used. Here, the Bayesian filter approach was presented as an approximation of a white box model, which embeds a strong prior belief on the description of the physics, albeit allowing for some modeling and measurement errors, typically (but not necessarily) assumed Gaussian. This results in the physics prescription imposed being highly strict, in that it defines the dynamics model within its specification. Comparatively, the *physics-guided* neural network also embeds a relatively high reliance on the physics-based model prescription, however, the NN allows for freedom in the estimation of the model output, akin to a residual modeling scheme. Comparing the two light-gray methods discussed in this paper, namely; ML-enhanced Bayesian filtering and physics-guided neural networks, both are often used when supplementary knowledge is required, but most of the underlying physics is well described with prior knowledge. Naturally, the flexibility of these approaches is relatively low, a facet sacrificed in order to improve precision in the physical descriptions. ML-enhanced BF techniques are often applied as estimators, for example, serving for the purpose of virtual sensing, within a system identification context. *PgNNs*, on the other hand, often aim to determine improved estimates of the measured output (e.g., displacement, velocity).

Light-gray methods often have shared motivations and characteristics as gray methods, but the gray approaches contain a higher degree of flexibility in terms of the embedded belief. In the case of “darker” such schemes, the model is still defined with a certain specificity, but they inherit greater flexibility owing to the more dominant incorporation of an ML method. However, such a lifting of restrictions stems from a different motivation for each approach. Dictionary methods allow for greater flexibility as they allow to combine several possible model forms, whereas the flexibility of *PINNs* may be attributed to the use of weak-form boundary conditions. Thus, in the former class, the embedded belief is essentially summarized in that a sparse representation of the defined dictionary will exhaustively describe the physics of the true system. For *PINNs*, the prior belief can be described as a reasonably accurate estimate of the model, but with some discrepancy resulting from either uncertainty in the governing parameters, or in the boundary conditions specification.

Dark-gray approaches offer maximal flexibility, that is, maximal potential to deviate from prior assumptions, among the *physics-enhanced* examples shown here. For such models, the embedded belief can be translated into regarding prior knowledge as a rather loose description of the *type* of system handled, or the *class/family* to which it belongs. Thus, such models are useful for improved generalisability, if this is a primary objective of the scheme, but also require a reasonably large amount of data to determine adequate estimates of the system. Inadequate levels of data may leave the system underfit, similar to issues with black-box models. However, in comparison to black-box models, the encoding the family of the system will allow for better interpolation, or potentially extrapolation, of the model output.

So far, most of the discussion has been centered around the flexibility and the beliefs embedded, and the fundamental facets the model aims to learn (model output or system parameters). But another aspect to consider of these methods is their enhanced-model structure, which can be defined as either unified or superimposed architectures. For a unified architecture, the model itself (e.g., the network or the kernels) will contain both the machine learning procedure, and the physics embedded. In the case of superimposed methods, the ML and the physics models are separate and the output of the model is formed by some combination of these two. This characteristic is not as conveniently correlated to a specific location within the spectrum plotted in [Figure 1](#) as is the aspect of flexibility. The dictionary, *PINN*, and *physics-encoded neural network* (PeNN) approaches can all be considered to be inherently unified models, whilst *PgNN* techniques are naturally superimposed models. However, depending on the specific type of approach, constrained GP and ML-enhanced filtering techniques can be either unified or superimposed architectures.

9. Conclusions and looking to the future

This paper has discussed, exhibited, and surveyed the spectrum of *PEML*, using the varied attributes of different methods to define and characterize them with respect to such a spectrum. This was done via a survey of recent applications/development of *PEML* within the wide field of structural mechanics, and through further demonstration of the alternate schemes on a simple running example of a Duffing oscillator. The motivation for, and application of, each of these variants will strongly depend on the use case, and the discussion and detailing of these methods in this paper should help not only the implementation of these techniques, but also for further research using an understanding of the—almost—philosophical implications of each method.

As we look toward the future of *PEML*, many pathways are opened in terms of development, understanding, and improvement. An existing challenge in machine learning techniques is to overcome the difficulties that manifest as a result of the increased dimensionality of problems. This challenge is far from circumvented in *PEML*; in fact, it becomes potentially more prominent in that *PEML* forms a compound of computational paradigms (both physics-based and data-driven), for each of which, dimensionality has a strong influence on the difficulty of implementation. One of the biggest advantages of informed ML, in comparison to black-box approaches, is their potential for improved inferability. However, the interpretability of the model is more difficult to immediately receive, and so further development can be done to improve this aspect, by utilizing domain transforms.

In order to improve the perception and utilization of *PEML* techniques, the unification of architecture styles, particularly for similar problems, is an invaluable development pathway. This could be to provide general design approaches given *PEML* design axioms, or to develop software packages or tools. Continuing on the technical development of *PEML* techniques, much like with black-box ML techniques, work is required to improve computing efficiency, especially at a time where reduction of energy consumption is increasingly important.

This paper has focused on *PEML* for structural mechanics, but one could also call attention to the potentially large impact of *PEML* as a natural next step in a society that is increasingly adopting, or opposing, AI. There are clear advantages of improved efficiency, lesser data requirements, and better generalisability. A highly impactful societal benefit may arise from the improvement of public trust in ML/AI when using informed models, since the opacity of ML models, and the lack of inferability, form a key contributor to public distrust (Toreini et al., 2020). The techno-societal study of the potential for improved public trust on the basis of *PEML*, would also provide an invaluable knowledge source for modern engineers and researchers, who can leverage such a knowledge for the development of tools with high societal impact.

Data availability statement. The code for the working example shown throughout this paper, including the “ground truth” simulator, is provided in a GitHub repository at <https://github.com/ETH-IBK-SMECH/PIDyNN>, or at the ETH Research Collection (DOI: 10.3929/ethz-b-000683329). The code is freely available and is written in Python, the specific requirements are provided in the repository.

Acknowledgments. The research was conducted as part of the Future Resilient Systems (FRS) program at the Singapore-ETH Centre, which was established collaboratively between ETH Zurich and the National Research Foundation Singapore. The authors gratefully acknowledge the funding from the Swiss National Science Foundation (SNSF) under the Horizon Europe funding guarantee, for the project “ReCharged—Climate-aware Resilience for Sustainable Critical and interdependent Infrastructure Systems enhanced by emerging Digital Technologies” (grant agreement No: 101086413).

Author contribution. Conceptualization: M.H.-A., W.L., K.B., E.C.; Methodology: M.H.-A., W.L., K.B., Z.L., E.C.; Software: M.H.-A., W.L., K.B., Z.L., E.C.; Investigation: M.H.A., W.L., K.B., Z.L., E.C.; Resources: E.C.; Data curation: M.H.-A., W.L., K.B., Z.L.; Writing—original draft: M.H.A., W.L., K.B., Z.L., E.C.; Writing—review and editing: M.H.-A., W.L., K.B., Z.L., E.C.; Supervision: Z.L., E.C.; Project administration: E.C.; Funding acquisition: E.C. All authors approved the final draft.

Funding statement. This research was supported by the National Research Foundation, Prime Minister’s Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) program. This work received no specific grant from any funding agency, commercial or not-for-profit sectors.

Competing interest. The authors declare none.

Ethical standard. The research meets all ethical guidelines, including adherence to the legal requirements of the study country.

References

- Abdessalem AB, Dervilis N, Wagg D and Worden K** (2018) Model selection and parameter estimation in structural dynamics using approximate Bayesian computation. *Mechanical Systems and Signal Processing* 99, 306–325.
- Abueidda DW, Koric S, Al-Rub RA, Parrott CM, James KA and Sobh NA** (2022) A deep learning energy method for hyperelasticity and viscoelasticity. *European Journal of Mechanics-A/Solids* 95, 104639.
- Abueidda DW, Lu Q and Koric S** (2021) Meshless physics-informed deep learning method for three-dimensional solid mechanics. *International Journal for Numerical Methods in Engineering* 122(23), 7182–7201.
- Angeli A, Desmet W and Naets F** (2021) Deep learning of multibody minimal coordinates for state and input estimation with Kalman filtering. *Multibody System Dynamics* 53(2), 205–223.
- Aucejo M, De Smet O and Deü J-F** (2019) Practical issues on the applicability of Kalman filtering for reconstructing mechanical sources in structural dynamics. *Journal of Sound and Vibration* 442, 45–70.
- Avendaño-Valencia LD, Chatzi EN, Koo KY and Brownjohn JM** (2017) Gaussian process time-series models for structures under operational variability. *Frontiers in Built Environment* 3, 69.
- Bacsa K, Lai Z, Liu W, Todd M and Chatzi E** (2023) Symplectic encoders for physics-constrained variational dynamics inference. *Scientific Reports* 13(1), 2643.
- Bayer J and Osendorfer C** (2015) Learning stochastic recurrent networks.
- Bergen KJ, Johnson PA, de Hoop MV and Beroza GC** (2019) Machine learning for data-driven discovery in solid earth geoscience. *Science* 363(6433), eaau0323.

- Bronstein MM, Bruna J, Cohen T and Veličković P** (2021) Geometric deep learning: Grids, groups, graphs, geodesics, and gauges.
- Brunton SL and Kutz JN** (2022) *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge: Cambridge University Press.
- Brunton SL, Proctor JL and Kutz JN** (2016a) Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences* 113(15), 3932–3937.
- Brunton SL, Proctor JL and Kutz JN** (2016b) Sparse identification of nonlinear dynamics with control (SINDYc). *IFAC-PapersOnLine* 49(18), 710–715.
- Chatzi EN and Smyth AW** (2009) The unscented Kalman filter and particle filter methods for nonlinear structural system identification with non-collocated heterogeneous sensing. *Structural Control and Health Monitoring* 16(1), 99–123.
- Chatzi EN, Smyth AW and Masri SF** (2010) Experimental application of on-line parametric identification for nonlinear hysteretic systems with model uncertainty. *Structural Safety* 32(5), 326–337. Probabilistic methods for modeling, simulation and optimization of engineering structures under uncertainty in honor of Jim Beck's 60th birthday.
- Chatzis MN, Chatzi EN and Smyth AW** (2015) On the observability and identifiability of nonlinear structural and mechanical systems. *Structural Control and Health Monitoring* 22(3), 574–593.
- Chen J and Liu Y** (2021) Probabilistic physics-guided machine learning for fatigue data analysis. *Expert Systems with Applications* 168, 114316.
- Chen RT, Rubanova Y, Bettencourt J and Duvenaud DK** (2018) Neural ordinary differential equations. *Advances in Neural Information Processing Systems* 31.
- Chen Z, Zhang J, Arjovsky M and Bottou L** (2020) Symplectic recurrent neural networks.
- Choudhary A, Lindner JF, Holliday EG, Miller ST, Sinha S and Ditto WL** (2020) Physics-enhanced neural networks learn order and chaos. *Physical Review E* 101(6), 062207.
- Christodoulou K and Papadimitriou C** (2007) Structural identification based on optimally weighted modal residuals. *Mechanical Systems and Signal Processing* 21(1), 4–23.
- Cross EJ, Gibson SJ, Jones MR, Pitchforth DJ, Zhang S and Rogers TJ** (2022) Physics-informed machine learning for structural health monitoring. In *Structural Health Monitoring Based on Data Science Techniques*. Cham: Springer, pp. 347–367.
- Cross E, Koo K, Brownjohn J and Worden K** (2013) Long-term monitoring and data analysis of the Tamar bridge. *Mechanical Systems and Signal Processing* 35(1–2), 16–34.
- Cross EJ and Rogers TJ** (2021) Physics-derived covariance functions for machine learning in structural dynamics. *IFAC-PapersOnLine* 54(7), 168–173.
- Cross EJ, Worden K and Chen Q** (2011) 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.
- Cuomo S, Di Cola VS, Giampaolo F, Rozza G, Raissi M and Piccialli F** (2022) Scientific machine learning through physics-informed neural networks: Where we are and what's next. *Journal of Scientific Computing* 92(3), 88.
- D'Amico B, Myers RJ, Sykes J, Voss E, Cousins-Jenvey B, Fawcett W, Richardson S, Kermani A and Pomponi F** (2019) Machine learning for sustainable structures: A call for data. In *Structures*, vol. 19. Amsterdam: Elsevier, pp. 1–4.
- Dardeno TA, Haywood-Alexander M, Mills RS, Bull LA, Dervilis N and Worden K** (2021) Investigating the effects of ambient temperature on feature consistency in vibration-based SHM. In *International Workshop on Structural Health Monitoring 2021*.
- David M and Méhats F** (2021) Symplectic learning for Hamiltonian neural networks.
- Dertimanis VK, Chatzi E, Azam SE and Papadimitriou C** (2019) Input-state-parameter estimation of structural systems from limited output information. *Mechanical Systems and Signal Processing* 126, 711–746.
- Desai SA, Mattheakis M, Sondak D, Protopapas P and Roberts SJ** (2021) Port-Hamiltonian neural networks for learning explicit time-dependent dynamical systems. *Physical Review E* 104(3), 034312.
- Dhadphale JM, Unni VR, Saha A and Sujith R** (2022) Neural ode to model and prognose thermoacoustic instability. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 32(1), 013131.
- Dou B, Zhu Z, Merkurjev E, Ke L, Chen L, Jiang J, Zhu Y, Liu J, Zhang B and Wei G-W** (2023) Machine learning methods for small data challenges in molecular science. *Chemical Reviews* 123(13), 8736–8780.
- Duenas-Osorio L and Vemuru SM** (2009) Cascading failures in complex infrastructure systems. *Structural Safety* 31(2), 157–167.
- Eftekhar Azam S, Chatzi E and Papadimitriou C** (2015) A dual Kalman filter approach for state estimation via output-only acceleration measurements. *Mechanical Systems and Signal Processing* 60–61, 866–886.
- Erazo K, Sen D, Nagarajaiah S and Sun L** (2019) Vibration-based structural health monitoring under changing environmental conditions using Kalman filtering. *Mechanical Systems and Signal Processing* 117, 1–15.
- Faroughi SA, Pawar N, Fernandes C, Das S, Kalantari NK and Mahjour SK** (2022) Physics-guided, physics-informed, and physics-encoded neural networks in scientific computing. arXiv preprint [arXiv:2211.07377](https://arxiv.org/abs/2211.07377).
- Farrar CR and Worden K** (2007) An introduction to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 365(1851), 303–315.
- Farrar CR and Worden K** (2012) *Structural Health Monitoring: A Machine Learning Perspective*. Hoboken, NJ: John Wiley & Sons.
- Feng W, Li Q and Lu Q** (2020) Force localization and reconstruction based on a novel sparse Kalman filter. *Mechanical Systems and Signal Processing* 144, 106890.

- Figueiredo E, Park G, Farrar CR, Worden K and Figueiras J** (2011) Machine learning algorithms for damage detection under operational and environmental variability. *Structural Health Monitoring* 10(6), 559–572.
- Flaschel M, Kumar S and De Lorenzis L** (2021) Unsupervised discovery of interpretable hyperelastic constitutive laws. *Computer Methods in Applied Mechanics and Engineering* 381, 113852.
- Frangos M, Marzouk Y, Willcox K and van Bloemen Waanders B** (2010) Surrogate and reduced-order modeling: A comparison of approaches for large-scale statistical inverse problems. In *Large-Scale Inverse Problems and Quantification of Uncertainty*. Hoboken, NJ: Wiley, pp. 123–149.
- Fuentes R, Nayek R, Gardner P, Dervilis N, Rogers T, Worden K and Cross E** (2021) Equation discovery for nonlinear dynamical systems: A Bayesian viewpoint. *Mechanical Systems and Signal Processing* 154, 107528.
- Geyer P, Singh MM and Chen X** (2021) Explainable AI for engineering design: A unified approach of systems engineering and component-based deep learning. arXiv preprint [arXiv:2108.13836](https://arxiv.org/abs/2108.13836).
- Goswami S, Anitescu C, Chakraborty S and Rabczuk T** (2020) Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. *Theoretical and Applied Fracture Mechanics* 106, 102447.
- Goyal P and Benner P** (2022) Discovery of nonlinear dynamical systems using a Runge–Kutta inspired dictionary-based sparse regression approach. *Proceedings of the Royal Society A* 478(2262), 20210883.
- Gréš S, Döhler M, Andersen P and Mevel L** (2021) Kalman filter-based subspace identification for operational modal analysis under unmeasured periodic excitation. *Mechanical Systems and Signal Processing* 146, 106996.
- Greydanus S, Dzamba M and Yosinski J** (2019) Hamiltonian neural networks. *CoRR*, abs/1906.01563.
- Grossmann TG, Komorowska UJ, Latz J and Schönlieb C-B** (2023) Can physics-informed neural networks beat the finite element method? arXiv preprint [arXiv:2302.04107](https://arxiv.org/abs/2302.04107).
- Guo K, Yang Z, Yu C-H and Buehler MJ** (2021) Artificial intelligence and machine learning in design of mechanical materials. *Materials Horizons* 8(4), 1153–1172.
- Haghighat E, Bekar AC, Madenci E and Juanes R** (2021a) Deep learning for solution and inversion of structural mechanics and vibrations. In *Modeling and Computation in Vibration Problems, Volume 2: Soft Computing and Uncertainty*. Bristol: IOP Publishing.
- Haghighat E, Raissi M, Moure A, Gomez H and Juanes R** (2021b) A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. *Computer Methods in Applied Mechanics and Engineering* 379, 113741.
- Haywood-Alexander M, Dervilis N, Worden K, Cross EJ, Mills RS and Rogers TJ** (2021) Structured machine learning tools for modelling characteristics of guided waves. *Mechanical Systems and Signal Processing* 156, 107628.
- He K, Zhang X, Ren S and Sun J** (2015) Deep residual learning for image recognition.
- Henkes A, Wessels H and Mahnken R** (2022) Physics informed neural networks for continuum micromechanics. *Computer Methods in Applied Mechanics and Engineering* 393, 114790.
- Hey T, Butler K, Jackson S and Thiyagalingam J** (2020) Machine learning and big scientific data. *Philosophical Transactions of the Royal Society A* 378(2166), 20190054.
- Huang D, Fuhg JN, Weißenfels C and Wriggers P** (2020) A machine learning based plasticity model using proper orthogonal decomposition. *Computer Methods in Applied Mechanics and Engineering* 365, 113008.
- Huang Z, Yin X and Liu Y** (2022) Physics-guided deep neural network for structural damage identification. *Ocean Engineering* 260, 112073.
- Jia X, Willard J, Karpatne A, Read J, Zwart J, Steinbach M and Kumar V** (2019) Physics guided RNNs for modeling dynamical systems: A case study in simulating lake temperature profiles. In *Proceedings of the 2019 SIAM International Conference on Data Mining*. Philadelphia, PA: SIAM, pp. 558–566.
- Jones MR, Rogers TJ and Cross EJ** (2023) Constraining Gaussian processes for physics-informed acoustic emission mapping. *Mechanical Systems and Signal Processing* 188, 109984.
- Jordan MI and Mitchell TM** (2015) Machine learning: Trends, perspectives, and prospects. *Science* 349(6245), 255–260.
- Joshi A, Thakolkaran P, Zheng Y, Escande M, Flaschel M, De Lorenzis L and Kumar S** (2022) Bayesian-EUCLID: Discovering hyperelastic material laws with uncertainties. *Computer Methods in Applied Mechanics and Engineering* 398, 115225.
- Kaiser E, Kutz JN and Brunton SL** (2018) Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proceedings of the Royal Society A* 474(2219), 20180335.
- Kamariotis A, Sardi L, Papaioannou I, Chatzi E and Straub D** (2023) On off-line and on-line Bayesian filtering for uncertainty quantification of structural deterioration. *Data-Centric Engineering* 4, e17.
- Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S and Yang L** (2021) Physics-informed machine learning. *Nature Reviews Physics* 3(6), 422–440.
- Karpatne A, Watkins W, Read J and Kumar V** (2017) Physics-guided neural networks (pgnn): An application in lake temperature modeling. arXiv preprint [arXiv:1710.11431](https://arxiv.org/abs/1710.11431), 2.
- Kim J-H, Jin J-W, Lee J-H and Kang K-W** (2017) Failure analysis for vibration-based energy harvester utilized in high-speed railroad vehicle. *Engineering Failure Analysis* 73, 85–96.
- Kingma DP and Ba J** (2014) Adam: A method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
- Kontoroupi T and Smyth AW** (2016) Online noise identification for joint state and parameter estimation of nonlinear systems. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering* 2(3), B4015006.

- Krishnan RG, Shalit U and Sontag D** (2016) Structured inference networks for nonlinear state space models. *arXiv:1609.09869* [cs, stat]. arXiv: 1609.09869.
- Lai Z, Liu W, Jian X, Bacsa K, Sun L and Chatzi E** (2022) Neural modal ordinary differential equations: Integrating physics-based modeling with neural ordinary differential equations for modeling high-dimensional monitored structures. *Data-Centric Engineering* 3, e34.
- Lai Z, Mylonas C, Nagarajaiah S and Chatzi E** (2021) Structural identification with physics-informed neural ordinary differential equations. *Journal of Sound and Vibration* 508, 116196.
- Langley P** (1994) Selection of relevant features in machine learning. *Proceedings of the AAAI Fall Symposium on Relevance* 184, 245–271.
- Li W, Bazant MZ and Zhu J** (2021) A physics-guided neural network framework for elastic plates: Comparison of governing equations-based and energy-based approaches. *Computer Methods in Applied Mechanics and Engineering* 383, 113933.
- Linardatos P, Papastefanopoulos V and Kotsiantis S** (2020) Explainable AI: A review of machine learning interpretability methods. *Entropy* 23(1), 18.
- Liu W, Lai Z, Bacsa K and Chatzi E** (2022) Physics-guided deep markov models for learning nonlinear dynamical systems with uncertainty. *Mechanical Systems and Signal Processing* 178, 109276.
- Liu W, Lai Z and Chatzi E** (2021) A physics-guided deep learning approach to modeling nonlinear dynamics: A case study of a Bouc-Wen system. *Structural Health Monitoring* 2021.
- Liu DC and Nocedal J** (1989) On the limited memory BFGS method for large scale optimization. *Mathematical Programming* 45 (1–3), 503–528.
- Lynch JP** (2007) An overview of wireless structural health monitoring for civil structures. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 365(1851), 345–372.
- Maes K, Chatzis M and Lombaert G** (2019) Observability of nonlinear systems with unmeasured inputs. *Mechanical Systems and Signal Processing* 130, 378–394.
- Maes K, Gillijns S and Lombaert G** (2018) A smoothing algorithm for joint input-state estimation in structural dynamics. *Mechanical Systems and Signal Processing* 98, 292–309.
- Molnar C** (2020) *Interpretable Machine Learning*. Lulu. com.
- Moradi S, Duran B, Eftekhar Azam S and Mofid M** (2023) Novel physics-informed artificial neural network architectures for system and input identification of structural dynamics PDEs. *Buildings* 13(3), 650.
- Muralidhar N, Bu J, Cao Z, He L, Ramakrishnan N, Tafti D and Karpadne A** (2020) Phynet: Physics guided neural networks for particle drag force prediction in assembly. In *Proceedings of the 2020 SIAM International Conference on Data Mining*. Philadelphia, PA: SIAM, pp. 559–567.
- Mylonas C, Abdallah I and Chatzi E** (2021) Conditional variational autoencoders for probabilistic wind turbine blade fatigue estimation using supervisory, control, and data acquisition data. *Wind Energy* 24(10), 1122–1139.
- Naets F, Croes J and Desmet W** (2015) An online coupled state/input/parameter estimation approach for structural dynamics. *Computer Methods in Applied Mechanics and Engineering* 283, 1167–1188.
- Nandakumar P and Jacob J** (2021) Structural and crack parameter identification on structures using observer Kalman filter identification/eigen system realization algorithm. *Journal of Solid Mechanics* 13(1), 68–79.
- Nandi T, Hennigh O, Nabian M, Liu Y, Woo M, Jordan T, Shahnam M, Syamlal M, Guenther C and VanEssendelft D** (2021) *Progress towards Solving High Reynolds Number Reacting Flows in Simnet*. Technical Report. Pittsburgh, PA: National Energy Technology Laboratory (NETL).
- Naser M** (2021) An engineer’s guide to eXplainable artificial intelligence and interpretable machine learning: Navigating causality, forced goodness, and the false perception of inference. *Automation in Construction* 129, 103821.
- Nayek R, Chakraborty S and Narasimhan S** (2019) A Gaussian process latent force model for joint input-state estimation in linear structural systems. *Mechanical Systems and Signal Processing* 128, 497–530.
- Nayek R, Fuentes R, Worden K and Cross EJ** (2021) On spike-and-slab priors for Bayesian equation discovery of nonlinear dynamical systems via sparse linear regression. *Mechanical Systems and Signal Processing* 161, 107986.
- Núñez M, Barreiro NL, Barrio RA and Rackauckas C** (2023) Forecasting virus outbreaks with social media data via neural ordinary differential equations. *Scientific Reports* 13(1), 10870.
- O’Driscoll P, Lee J and Fu B** (2019) Physics enhanced artificial intelligence. arXiv preprint arXiv:1903.04442.
- O’Hagan A** (1978) Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society: Series B (Methodological)* 40(1), 1–24.
- Odelson BJ, Rajamani MR and Rawlings JB** (2006) A new autocovariance least-squares method for estimating noise covariances. *Automatica* 42(2), 303–308.
- Olson RS, Cava WL, Mustahsan Z, Varik A and Moore JH** (2018) Data-driven advice for applying machine learning to bioinformatics problems. In *Pacific Symposium on Biocomputing 2018: Proceedings of the Pacific Symposium*. Singapore: World Scientific, pp. 192–203.
- Padonou E and Roustant O** (2016) Polar Gaussian processes and experimental designs in circular domains. Working paper or preprint.
- Papadimitriou C and Katafygiotis LS** (2004) Bayesian modeling and updating. In *Engineering Design Reliability Handbook*. Boca Raton, FL: CRC Press, pp. 525–544.
- Park T and Casella G** (2008) The Bayesian lasso. *Journal of the American Statistical Association* 103(482), 681–686.

- Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Kopf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J and Chintala S (2019) PyTorch: An imperative style, high-performance deep learning library. In Wallach H, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox E and Garnett R (eds.), *Advances in Neural Information Processing Systems* 32. New York: Curran Associates, Inc, pp. 8024–8035.
- Pawar S, San O, Nair A, Rasheed A and Kvamsdal T (2021) Model fusion with physics-guided machine learning: Projection-based reduced-order modeling. *Physics of Fluids* 33(6), 067123.
- Peeters B and De Roeck G (2001) Stochastic system identification for operational modal analysis: A review. *Journal Dynamic Systems, Measurements, and Control* 123(4), 659–667.
- Petersen Ø, Øiseth O and Lourens E (2022) Wind load estimation and virtual sensing in long-span suspension bridges using physics-informed Gaussian process latent force models. *Mechanical Systems and Signal Processing* 170, 108742.
- Pontryagin LS, Mishchenko E, Boltyanskii V and Gamkrelidze R (1962) *The Mathematical Theory of Optimal Processes*. Hoboken, NJ: Wiley.
- Raissi M, Perdikaris P and Karniadakis GE (2019) 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.
- Rebillat M, Monteiro E and Mechbal N (2023) Physically informed and data driven direct models for lamb waves based shm: Advantages and drawbacks of existing approaches. In *Proceedings of the 14th International Workshop on Structural Health Monitoring (IWSHM 2024)*. Stanford: IWSHM, pp. 645–652.
- Reich Y (1997) Machine learning techniques for civil engineering problems. *Computer-Aided Civil and Infrastructure Engineering* 12(4), 295–310.
- Reimann D, Nidadavolu K, Hassan H, Vajragupta N, Glasmachers T, Junker P and Hartmaier A (2019) Modeling macroscopic material behavior with machine learning algorithms trained by micromechanical simulations. *Frontiers in Materials* 6, 181.
- Ren Z, Han X, Yu X, Skjetne R, Leira BJ, Sævik S and Zhu M (2023) Data-driven simultaneous identification of the 6DOF dynamic model and wave load for a ship in waves. *Mechanical Systems and Signal Processing* 184, 109422.
- Revach G, Shlezinger N, Ni X, Escoriza AL, Van Sloun RJ and Eldar YC (2022) KalmanNet: Neural network aided Kalman filtering for partially known dynamics. *IEEE Transactions on Signal Processing* 70, 1532–1547.
- Rezaei S, Harandi A, Moeineddin A, Xu B-X and Reese S (2022) 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.
- Ritto T and Rochinha F (2021) Digital twin, physics-based model, and machine learning applied to damage detection in structures. *Mechanical Systems and Signal Processing* 155, 107614.
- Robinson H, Pawar S, Rasheed A and San O (2022) Physics guided neural networks for modelling of non-linear dynamics. *Neural Networks* 154, 333–345.
- Rogers T, Worden K and Cross E (2020) On the application of Gaussian process latent force models for joint input-state-parameter estimation: With a view to Bayesian operational identification. *Mechanical Systems and Signal Processing* 140, 106580.
- Saemundsson S, Terenin A, Hofmann K and Deisenroth MP (2020) Variational integrator networks for physically structured embeddings.
- Sanchez-Gonzalez A, Bapst V, Cranmer K and Battaglia P (2019) Hamiltonian graph networks with ODE integrators.
- Sedehi O, Papadimitriou C, Teymouri D and Katafygiotis LS (2019) Sequential Bayesian estimation of state and input in dynamical systems using output-only measurements. *Mechanical Systems and Signal Processing* 131, 659–688.
- Shadabfar M, Mahsuli M, Zhang Y, Xue Y, Ayyub BM, Huang H and Medina RA (2022) Resilience-based design of infrastructure: Review of models, methodologies, and computational tools. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering* 8(1), 03121004.
- Shaikhina T, Lowe D, Daga S, Briggs D, Higgins R and Khovanova N (2015) Machine learning for predictive modelling based on small data in biomedical engineering. *IFAC-PapersOnLine* 48(20), 469–474.
- Sharifani K and Amini M (2023) Machine learning and deep learning: A review of methods and applications. *World Information Technology and Engineering Journal* 10(07), 3897–3904.
- Shi X and Chatzis M (2022) An efficient algorithm to test the observability of rational nonlinear systems with unmeasured inputs. *Mechanical Systems and Signal Processing* 165, 108345.
- Simpson T, Dervilis N and Chatzi E (2021) Machine learning approach to model order reduction of nonlinear systems via autoencoder and LSTM networks. *Journal of Engineering Mechanics* 147(10), 04021061.
- Simpson T, Tsioliamanis G, Dervilis N, Worden K and Chatzi E (2023) On the use of variational autoencoders for nonlinear modal analysis. In Brake MR, Renson L, Kuether RJ and Tiso P (eds.), *Nonlinear Structures & Systems, Vol. 1*. Cham: Springer International Publishing, pp. 297–300.
- Sohn H (2007) Effects of environmental and operational variability on structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 365(1851), 539–560.
- Sohn H, Farrar CR, Hemez FM, Shunk DD, Stinematos DW, Nadler BR and Czarnecki JJ (2003) A review of structural health monitoring literature: 1996–2001. *Los Alamos National Laboratory, USA* 1, 16.
- Solin A and Särkkä S (2020) Hilbert space methods for reduced-rank Gaussian process regression. *Statistics and Computing* 30(2), 419–446.

- Sosanya A and Greydanus S** (2022) Dissipative Hamiltonian neural networks: Learning dissipative and conservative dynamics separately. *CoRR*, abs/2201.10085.
- Sun H, Burton HV and Huang H** (2021) Machine learning applications for building structural design and performance assessment: State-of-the-art review. *Journal of Building Engineering* 33, 101816.
- Sun L, Gao H, Pan S and Wang J-X** (2020) Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Computer Methods in Applied Mechanics and Engineering* 361, 112732.
- Sun F, Liu Y, Wang Q and Sun H** (2023) PiSL: Physics-informed spline learning for data-driven identification of nonlinear dynamical systems. *Mechanical Systems and Signal Processing* 191, 110165.
- Tatsis K, Agathos K, Chatzi E and Dertimanis V** (2022) A hierarchical output-only Bayesian approach for online vibration-based crack detection using parametric reduced-order models. *Mechanical Systems and Signal Processing* 167, 108558.
- Tatsis K, Dertimanis V and Chatzi E** (2023) On off-line and on-line Bayesian filtering for uncertainty quantification of structural deterioration. *Journal of Structural Dynamics* 4, e17.
- Tatsis K, Dertimanis V, Papadimitriou C, Lourens E and Chatzi E** (2021) A general substructure-based framework for input-state estimation using limited output measurements. *Mechanical Systems and Signal Processing* 150, 107223.
- Tchemodanova SP, Sanayei M, Moaveni B, Tatsis K and Chatzi E** (2021) Strain predictions at unmeasured locations of a substructure using sparse response-only vibration measurements. *Journal of Civil Structural Health Monitoring* 11(4), 1113–1136.
- Teymouri D, Sedehi O, Katafygiotis LS and Papadimitriou C** (2023) Input-state-parameter-noise identification and virtual sensing in dynamical systems: A Bayesian expectation-maximization (BEM) perspective. *Mechanical Systems and Signal Processing* 185, 109758.
- Thakolkaran P, Joshi A, Zheng Y, Flaschel M, De Lorenzis L and Kumar S** (2022) NN-EUCLID: Deep-learning hyperelasticity without stress data. *Journal of the Mechanics and Physics of Solids* 169, 105076.
- Tibshirani R** (1996) Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 58(1), 267–288.
- Toreini E, Aitken M, Coopamootoo K, Elliott K, Zelaya CG and Van Moorsel A** (2020) The relationship between trust in ai and trustworthy machine learning technologies. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*. New York: ACM, pp. 272–283.
- Van der Meer F, Sluys L, Hallett S and Wisnom M** (2012) Computational modeling of complex failure mechanisms in laminates. *Journal of Composite Materials* 46(5), 603–623.
- Vettori S, Di Lorenzo E, Peeters B, Luczak M and Chatzi E** (2023a) An adaptive-noise augmented Kalman filter approach for input-state estimation in structural dynamics. *Mechanical Systems and Signal Processing* 184, 109654.
- Vettori S, Lorenzo ED, Peeters B and Chatzi E** (2023b) Assessment of alternative covariance functions for joint input-state estimation via Gaussian process latent force models in structural dynamics.
- Vincent P and Bengio Y** (2002) Kernel matching pursuit. *Machine Learning* 48, 165–187.
- Vlachas PR, Arampatzis G, Uhler C and Koumoutsakos P** (2022) Multiscale simulations of complex systems by learning their effective dynamics. *Nature Machine Intelligence* 4(4), 359–366.
- Wang S, Teng Y and Perdikaris P** (2021) Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing* 43(5), A3055–A3081.
- Williams CK and Rasmussen CE** (2006) *Gaussian Processes for Machine Learning*, Vol. 2. Cambridge, MA: MIT Press.
- Wu K and Xiu D** (2020) Data-driven deep learning of partial differential equations in modal space. *Journal of Computational Physics* 408, 109307.
- Xiaowei J, Shujin L and Hui L** (2021) Physics-enhanced deep learning methods for modelling and simulating flow fields. *Chinese Journal of Theoretical and Applied Physics* 53(10), 2616–2629.
- Xu P-F, Han C-B, Cheng H-X, Cheng C and Ge T** (2022) A physics-informed neural network for the prediction of unmanned surface vehicle dynamics. *Journal of Marine Science and Engineering* 10(2), 148.
- Yang Y, Nagayama T and Xue K** (2020) Structure system estimation under seismic excitation with an adaptive extended Kalman filter. *Journal of Sound and Vibration* 489, 115690.
- Yin X, Huang Z and Liu Y** (2023) Bridge damage identification under the moving vehicle loads based on the method of physics-guided deep neural networks. *Mechanical Systems and Signal Processing* 190, 110123.
- Yoshida H** (1990) Construction of higher order symplectic integrators. *Physics Letters A* 150(5), 262–268.
- Yu Y, Yao H and Liu Y** (2020) Structural dynamics simulation using a novel physics-guided machine learning method. *Engineering Applications of Artificial Intelligence* 96, 103947.
- Yuan F-G, Zargar SA, Chen Q and Wang S** (2020) Machine learning for structural health monitoring: Challenges and opportunities. *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2020* 11379, 1137903.
- Yucesan YA and Viana FA** (2020) A physics-informed neural network for wind turbine main bearing fatigue. *International Journal of Prognostics and Health Management* 11(1), 17.
- Zhang ZJ and Duraisamy K** (2015) Machine learning methods for data-driven turbulence modeling. In *22nd AIAA Computational Fluid Dynamics Conference*. Dallas, TX: AIAA, p. 2460.
- Zhang Y and Ling C** (2018) A strategy to apply machine learning to small datasets in materials science. *npj Computational Materials* 4(1), 25.

- Zhang S, Rogers TJ and Cross EJ** (2021) Gaussian process based grey-box modelling for SHM of structures under fluctuating environmental conditions. In *European Workshop on Structural Health Monitoring: Special Collection of 2020 Papers-Volume 2*. Cham: Springer, pp. 55–66.
- Zhang Z and Sun C** (2021) Structural damage identification via physics-guided machine learning: A methodology integrating pattern recognition with finite element model updating. *Structural Health Monitoring* 20(4), 1675–1688.
- Zhang E, Yin M and Karniadakis GE** (2020) Physics-informed neural networks for nonhomogeneous material identification in elasticity imaging. arXiv preprint [arXiv:2009.04525](https://arxiv.org/abs/2009.04525).
- Zheng B, Li T, Qi H, Gao L, Liu X and Yuan L** (2022) Physics-informed machine learning model for computational fracture of quasi-brittle materials without labelled data. *International Journal of Mechanical Sciences* 223, 107282.
- Zhong YD, Dey B and Chakraborty A** (2020) Symplectic ODE-Net: Learning hamiltonian dynamics with control.
- Zhong S, Zhang K, Bagheri M, Burken JG, Gu A, Li B, Ma X, Marrone BL, Ren ZJ, Schrier J, et al.** (2021) Machine learning: New ideas and tools in environmental science and engineering. *Environmental Science & Technology* 55(19), 12741–12754.
- Zhuang X, Guo H, Alajlan N, Zhu H and Rabczuk T** (2021) Deep autoencoder based energy method for the bending, vibration, and buckling analysis of kirchhoff plates with transfer learning. *European Journal of Mechanics-A/Solids* 87, 104225.
- Zou J, Lourens E-M and Cicirello A** (2023) Virtual sensing of subsoil strain response in monopile-based offshore wind turbines via Gaussian process latent force models. *Mechanical Systems and Signal Processing* 200, 110488.