Physics-Informed Neural Networks for One-Step-Ahead Prediction of Dynamical Systems

MARCUS HAYWOOD-ALEXANDER and ELENI CHATZI

ABSTRACT

During online implementation of vibration-based structural health monitoring (SHM) strategies, forward prediction of the system state may allow for improved detection speed. With adequately fast forward prediction, feedback systems can also be improved to provide in-time control plants. When using solely physics-driven models, small discrepancies between the physical system and digital model can result in significant deviation between the estimated and true output. On the opposite end, a solely data-driven model can only reasonably be applied in an identical, or sufficiently similar, scenario to that for which the data was collected. Using machine learning to combine data with known physics is a well-proved approach to overcoming this issue. One such method for this approach is the use of physics-informed neural networks (PINNs), which can be implemented as either a forward modeller, or a constrained learner, for equation solution discovery, or equation discovery. The former of these aims to provide the desired output from the governing physical equations, whereas the latter estimates the parameters in these equations. A significant advantage of PINNs is, given a suitable network architecture, the high speed and low computational-cost of their prediction step, which positions them as a useful approach for real-time estimation, given adequate training. A common assumption for PINNs is that the embedded physics is exhaustive with respect to the 'true' model. In this paper, a novel PINN-based architecture is presented to rapidly forward-predict the state of a dynamic system given an initial state. For the state estimation, the PINN acts as an equation *solution* discovery approach, and the novelty of the architecture here is to provide a more generalised predictor which can be applied to a wider range of instances. The PINN is intended to deliver its prediction within a prescribed time frame, which equals the sampling time of the acquisition/control system, and is here assessed against this goal.

Marcus Haywood-Alexander, Research Associate, Email: mhaywood@ethz.ch. Chair of Structural Mechanics and Monitoring, Institute of Structural Engineering, ETH Zürich, Zürich, Switzerland.

INTRODUCTION

Structural dynamics forms the main source of traits employed for detecting damage in a structural health monitoring regime [1], which often requires an accurate and robust model of the system for physics-only based models. Such features are commonly extracted from frequency-domain methods, such as modal characteristics, transmissibilities, which provide robust parameters which can be used in novelty-detection regimes [2]. For structures that vibrate at lower frequencies, these methods will require large sampling lengths, which fosters alternate adoption of time-domain models for improving detection speeds [3]. With adequately fast forward prediction, feedback systems may also be improved to provide in-time control plants [4].

Real-, or in-, time state estimation has had many proposed approaches, particularly in vehicle dynamics [5]. Kalman filters have been used to estimate the state of dynamic systems [6, 7], which, with an accurate model of the system, can produce reliable and reasonably fast estimation. However, they require the use of matrix operations, which are computationally of the order $\mathcal{O}(n^3)$, and so can quickly require long computational times. Parametric-reduced-order-modelling has been used to approach the problem of rapid estimation [8], where data-driven methods ar used to allow reconstruction in the time domain using a reduced set of parameters which retains a prescribed level of information. Solely data-based approaches have also shown promise for rapid identification of nonlinear dynamic systems [9], but such methods can suffer from a lack of generalisability.

Purely physics-drive modelling can be a sure way to obtain accurate models, assuming an adequate definition, however, significant errors between the estimated and true output can result from small discrepancies between this definition and the true physical system. This can motivate the use of solely data-driven modelling, but such methods suffer from a lack of generalisation; they may only reasonably be applied in an identical, or sufficiently similar, scenario to that for which the data was collected. By combining known physics with machine learning, an improved modelling strategy can be obtained. One such method is the *physics-informed neural network* (PINN) [10], which has gained attention in recent years thanks to its flexibility [11]; it can be implemented as a constrained learner for either equation solution discovery [12] or equation discovery [13], or as a forward modeller [14].

The training time of PINNs as forward modellers, in relation to traditional finite-element methods, is often much larger, leading to some statements implying their impracticality [15]. There have also been claims that PINNs are also slower in computation mode [16], however, these studies lack analysis on the minimum network architecture required, which is noted by Rezaei, et al [15]. As a result of the relatively low computational cost of a single-pass (with suitable network design), PINNs may provide a convenient solution for problems where rapid prediction is required [17], as the computational effort is done *a priori* during training. Furthermore, the inherent stochastic and iterative procedures involved with machine learning approaches, place them in a useful standing for uncertainty quantification; which is a valuable resource in SHM techniques [1].

In this work, a PINN is designed to predict, forward in time, the state of a dynamic system, given an initial state. A frequently-encountered challenge with PINNs for dynamic systems is the difficulty in estimating over a large horizon, both in time or space.

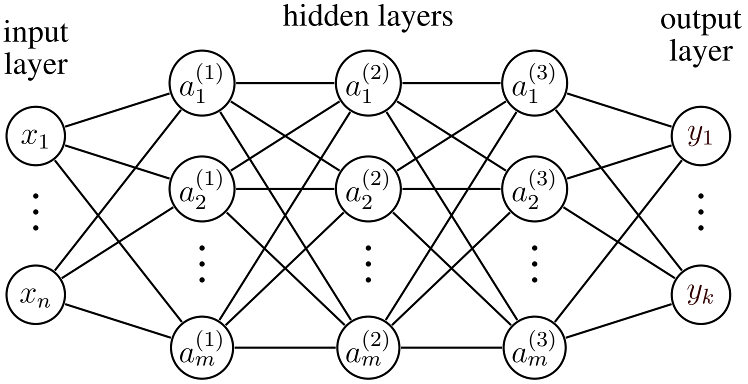

Figure 1. Example architecture of an arbitrary fully-connected, feed-forward neural network.

Therefore, many applications discretise the domain, applying separate NNs to each of these discretised portions [18]. The primary issue with such techniques, is their inflexibility; they are restricted to the exact instance in which they are trained. Many PINNs for dynamic systems provide solutions for a given initial condition [19], however, with even minor deviations from this scenario, the predictor is no longer suitable. The aim here is to determine a model that can predict within a specified time window, so it is not necessary to predict over such a large horizon, however, the discretisation is done similarly here.

PHYSICS-INFORMED NEURAL NETWORKS

Perhaps one of the most well-known machine learning methods, is the artificial neural network (ANN). For a regression problem, the aim of an ANN is to determine an estimate of the mapping from the input $\mathbf{x}$, to the output $\mathbf{y}$. An example of a fully-connected feed-forward NN architecture is shown in Figure 1, where each node is 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,

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

where $\mathbf{W} = \{\mathbf{w}^1, ..., \mathbf{w}^N\}$ and $\mathbf{B} = \{\mathbf{b}^1, ..., \mathbf{b}^N\}$ are the weights and biases of the network, respectively. These then form the hyperparameters of the network, $\mathbf{\Theta} = \{\mathbf{W}, \mathbf{B}\}$.

The aim of the training stage is to then determine the network parameters $\mathbf{\Theta} = \{\mathbf{W}, \mathbf{B}\}$, which is accomplished by minimising an objective function designed with an intention to be towards zero. With target output data $\mathbf{y}^*$ from the domain of observations $\mathbf{x} \in \Omega_0$, this objective function is often defined as the mean squared error between the predicted values and the observed values,

$$L_{obs}(\mathbf{x}^*;\mathbf{\Theta}) = \langle \mathbf{y}^* - \mathcal{N}_{\mathbf{y}}(\mathbf{x}^*;\mathbf{\Theta}) \rangle_{\Omega_0}, \qquad \langle \bullet \rangle_{\Omega_\kappa} = \frac{1}{N_\kappa} \sum_{x \in \Omega_\kappa} ||\bullet||^2 \tag{2}$$

If the physics of the system is known (or estimated) in the form of ordinary or partial differential equations, then this can be embedded into the objective function over which the NN parameters are optimised [10]. Given a general form of the PDE,

$$\mathcal{F}(\mathbf{y}, \mathbf{x}; \theta) = 0 \tag{3}$$

for some nonlinear operator $\mathcal{F}$ acting on $\mathbf{y}(\mathbf{x})$, where θ are parameters of the equation.

When predicting the output of a neural network, we can also create an estimate of the nonlinear operator, $\mathcal{F}(\mathcal{N}_{\mathbf{y}}, \mathbf{x}; \theta)$. This can then be directly used as an additional objective function, forming a 'soft' boundary condition [20], to be minimised, as when this term vanishes, the PDE is satisfied. Given the domain of collocation points, $\mathbf{x}_p \in \Omega_p$, this term is defined as,

$$L_{pde}(\mathbf{x}_p; \mathbf{\Theta}, \theta) = \langle \mathcal{F}(\mathcal{N}_{\mathbf{y_p}}, \mathbf{x}_p; \theta) \rangle_{\Omega_p}, \qquad \mathcal{N}_{\mathbf{y_p}} = \mathcal{N}_{\mathbf{y}}(\mathbf{x}_p; \mathbf{\Theta}) \tag{4}$$

Then, we can combine the observation objective function with the pde objective function, and minimise this,

$$L = L_{obs} + \Lambda L_{pde} \tag{5}$$

where Λ is a normalisation parameter required to posit the objective function terms in the same magnitude to aid optimisation. In this work, often a combination of the input normalisation parameters are used to set the value of Λ.

ONE-STEP-AHEAD PINN

The aim of the *one-step-ahead* predictor in this work, $\mathcal{N}_w$, is to estimate the displacement of a single-degree-of-freedom (SDOF) system within a given time horizon, from its initial state and the applied force,

$$w(t) = \mathcal{N}_w(w_0, \dot{w}_0, f_0, t; \mathbf{\Theta}) \tag{6}$$

where w_0 and $\dot{w}_0$ are the initial displacement and velocities, respectively, f_0 is the force and t is the time, where $w(t=0) = w_0$. Antonelo, et al. [17], proposed an architecture for rapid control estimation using a PINN-based approach. A similar architecture is proposed here, however, is adapted to include the velocity term so that it may better model nonlinear systems. Given a set of data $\mathbf{y}^* = \{w, \dot{w}\}$, the time-horizon T, equal to the sampling period of the data, is set as the target domain for the predictor. This data can then be set as the initial state inputs, $\{w_0, \dot{w}_0\}$, and the observation loss is then defined as,

$$L_{obs} = \langle \mathcal{N}_w - w^* \rangle_{\Omega \in t=T} \tag{7}$$

where w^* is taken from the sample one time point ahead, i.e. $w^{*,(k)} = w_0^{(k+1)}$. Then, the collocation domain is set as a uniform sample over the time-horizon, $\Omega_c \in [0, T]$, and the physics loss is defined as,

$$L_{pde} = \langle \mathcal{F}(\mathcal{N}_w) \rangle_{\Omega_c} \tag{8}$$

Additionally, an initial-condition loss was defined,

$$L_{ic} = \langle w_0 - \mathcal{N}_w \rangle_{\Omega \in t=0} + \langle \dot{w}_0 - \partial_t \mathcal{N}_w \rangle_{\Omega \in t=0} \tag{9}$$

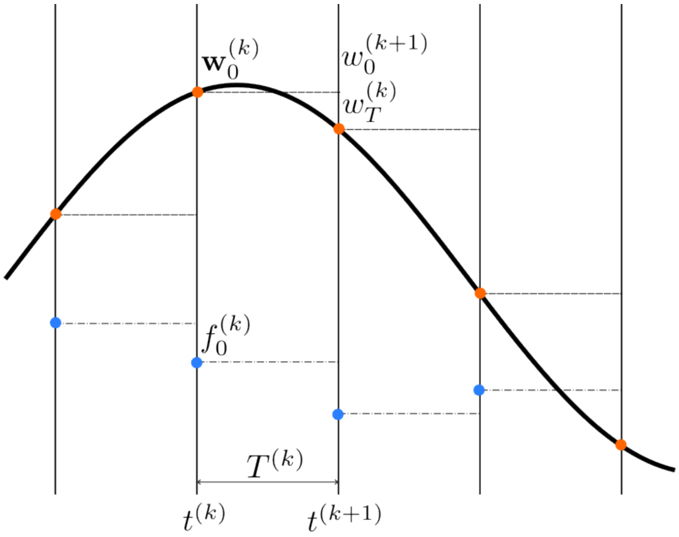

Figure 2. Data representation from a time-domain signal, indicating the initial state for data point k, and the data available at the end of the time horizon.

The role of each of these objective functions is to apply individual 'soft' conditions on the system. The first term guides the learner using observations at the end of a specified time horizon. The second term, constrains the learner using known physics, so the learner can produce solutions within the collocation domain where no observations took place. And the final term provides a constraint on the collocation domain at time zero as we are predicting *from* this state. In this work, a SDOF duffing oscillator, with force $f(t)$ applied, is modelled,

$$\mathcal{F} = m\ddot{w} + c\dot{w} + kw + k_3 w^3 - f(t) = 0 \tag{10}$$

As an overall description, this method acts as an equation solution discovery in the time horizon between samples. Because the PINN accepts only the initial state and force input, the method is readily available to make state predictions in a large number of instances (i.e. it is not limited to a specific initial condition), given adequate training.

Figure 2 shows how the problem is formulated in the time domain. The data point at location k is treated as the current system state $w_0^{(k)}$, and the aim is to predict the displacement at the end of the time-horizon, $w_T^{(k)}$. As such, the subsequent data point is taken as the observed displacement at the end of this window, i.e. $w_T^{(k)} = w_0^{(k+1)}$.

RESULTS AND DISCUSSION

The method detailed above was applied to a variety of scenarios for an SDOF system, where the data was simulated using Runga-Kutta numerical solutions [21]. Firstly, the system was set up as either a linear or Duffing oscillator, the former of which was done by setting $k_3 = 0$. Then, for each of these system definitions, either free or forced vibration was modelled. Four different forcing types were modelled; sinusoidal, random-phase-multi-sine (RPMS), Gaussian white noise (GWN), and forward-sine-sweep. Further details on the parameters of the systems being modelled are shown in Table I. In

TABLE I. MODEL PARAMETERS FOR DYNAMIC SYSTEMS AND DATA GENERATION.

	T	N	N_s	m	k	c	k_3
Free	0.3519 s	1024	127	10 kg	15 N/m	0.75 Ns/m	20 Nm^{-3}
Forced	0.4106 s	2048	511	10 kg	15 N/m	1.0 Ns/m	20 Nm^{-3}

all results in this work, a fully-connected, feed-forward neural-network was used with 2 hidden layers, each of 16 nodes. This network architecture was chosen based on the modelled system being relatively simple and the objective of enabling rapid estimation.

For all system/excitation types, an N sample dataset was generated which includes displacement, velocity and force with an equivalent sampling period T. This is then used to generate an $N-1$ sample dataset of,

$$\mathcal{D} = w^*(w_0, \dot{w}_0, f_0, T) \tag{11}$$

Then, the PINN architecture is trained on N_s samples from this dataset, selected using a quasi-random Sobol sequence [22]. After training, the remaining N_t samples were passed through the PINN to estimate the displacement at the end of the time horizon. As an assessment metric, the *root-mean-square-error* was calculated,

$$RMSE = \sqrt{\frac{1}{N_t}\sum_{i=1}^{N_t} |w^*(w_0, \dot{w}_0, f_0, T) - \mathcal{N}_w(w_0, \dot{w}_0, f_0, T)|^2} \tag{12}$$

Table II shows the assessment metrics for all the system/excitation types modelled, including the RMSE and the maximum computational time (MCT) for all *individual* predictions. The computation was performed on a MacBook Air with Apple M2 CPU.

For all predictions, the maximum computational time is much lower than the sampling periods. In particular, the time required for prediction of linear and nonlinear systems remains similar, which would not be the case for conventional numerical solutions. It is worth noting that the computational time required is dependent on the system, and the computational performance is dependent on hardware, and so these factors would be necessary to design for in application.

The RMSE metric provides information on the error of the predictions at the same magnitude as the value begin estimated. Therefore, the values in Table II indicate a good level of accuracy of the predictor. The largest value for the RMSE is the linear system undergoing RPMS excitation, and information on this prediction over the time domain

TABLE II. ASSESSMENT METRICS FROM PREDICTION USING TRAINED NEURAL NETWORKS.

Oscillator	Metric		Forced			
		Free	Sinusoid	RPMS	GWN	Sine sweep
Linear	RMSE	1.122E-4	3.268E-5	1.406E-3	1.348E-4	7.398E-4
	MCT (ns)	39.0	40.3	44.9	39.6	39.3
Duffing	RMSE	7.989E-5	3.280E-5	8.231E-4	1.656E-4	1.267E-4
	MCT (ns)	39.9	39.7	40.9	40.9	39.7

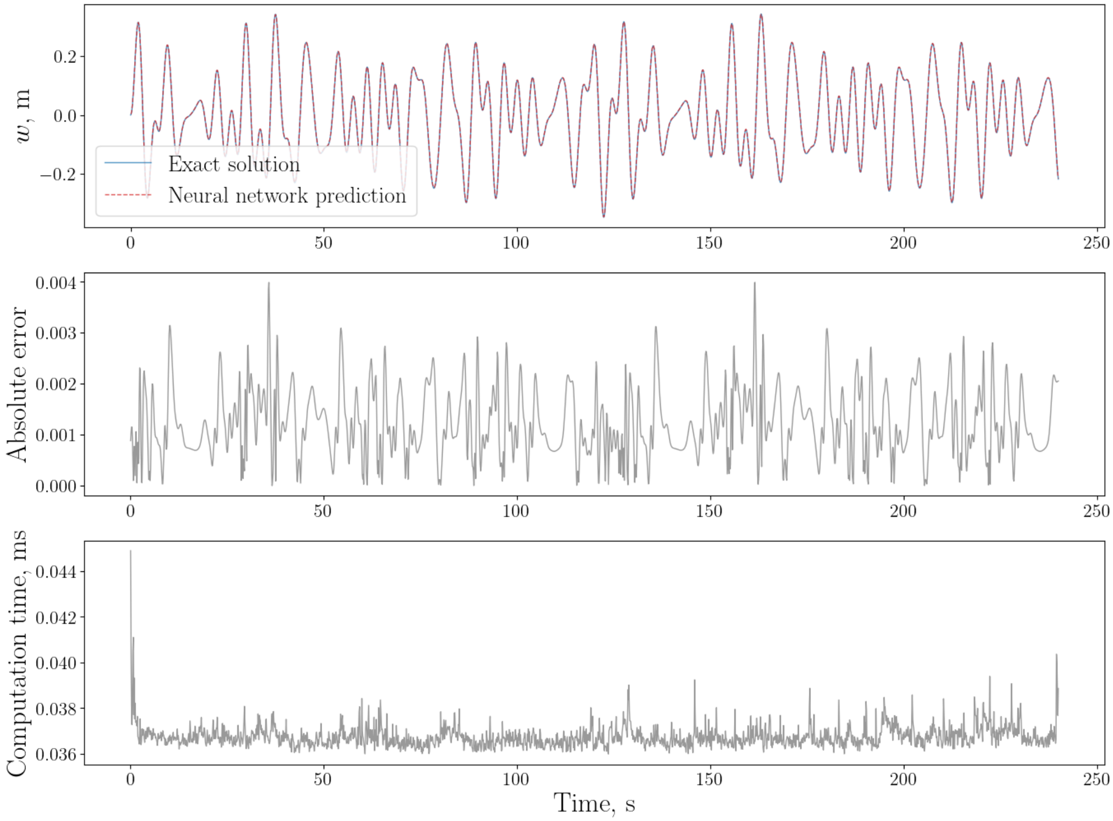

Figure 3. Results for prediction of a linear oscillator with RPMS excitation; (a) recreated signal by prediction at each initial state point extracted from the signal, (b) absolute error between estimated displacement and exact solution, and (c) computation time for each prediction step.

is shown in Figure 3. From this figure it can be seen that the RMSE values are, at most, approximately two order of magnitude smaller than the value being estimated.

Figure 3 and Figure 4 show the prediction of the trained PINN at every time step available; the full signal is recreated by taking the current state and predicting one time step ahead. As well as the recreated signal, the absolute error between the exact solution and the prediction, and the CPU time at each data point, are shown. Figure 3 shows the results for the prediction of the Duffing oscillator system undergoing random-phase-multi-sine excitation, and Figure 3 shows the same system undergoing Gaussian white noise excitation. Even though the system modelled is a nonlinear system, which requires iterative numerical solutions, the predictions are accurate throughout the entire domain.

The accuracy of the reconstructed signal also displays the methods capability as a forward modeller, which could be performed by feeding forward the results of the estimated solution of the previous system state to the next. As well as the accuracy in prediction, Figures 3 and 4 show the consistency in the computational speeds at all state prediction points, with a deviation only between ∼36 to 44 ns.

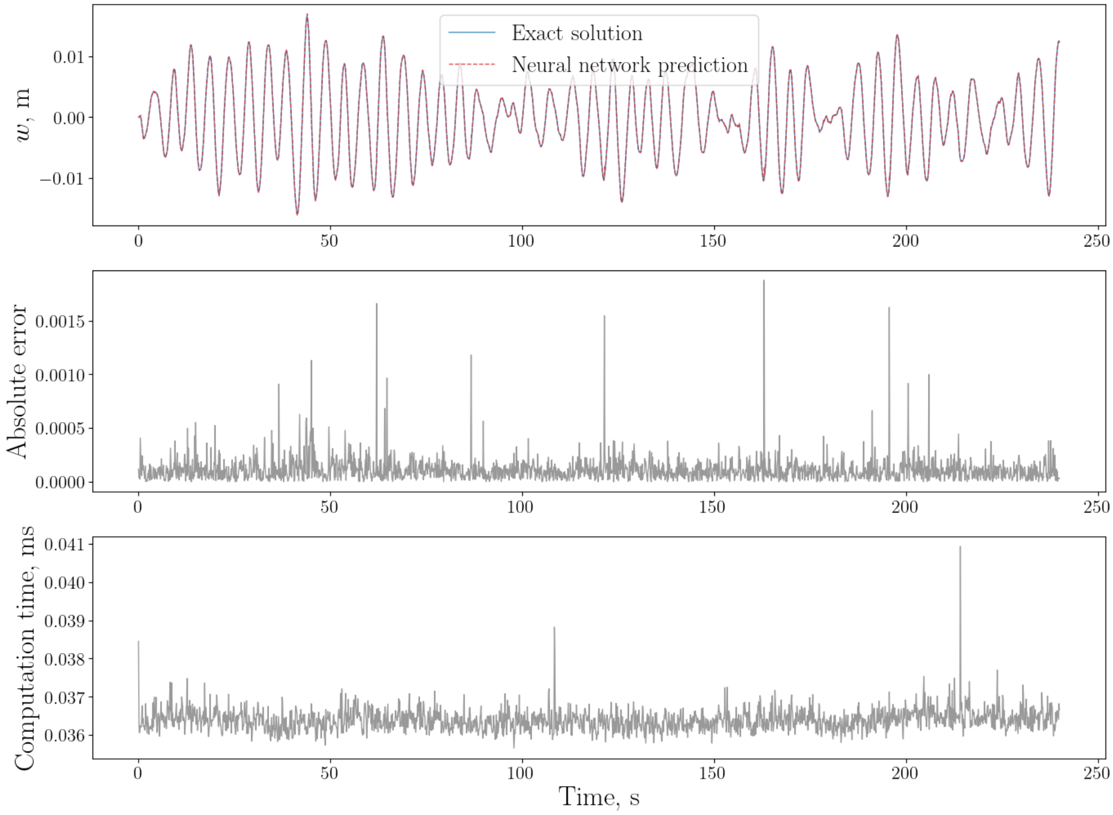

Figure 4. Results for prediction of a Duffing oscillator with GWN excitation; (a) recreated signal by prediction at each initial state point extracted from the signal, (b) absolute error between estimated displacement and exact solution, and (c) computation time for each prediction step.

CONCLUDING REMARKS

In this paper, a physics-informed-neural-network (PINN)-based approach for rapid forward prediction of a dynamic system was shown. A variety of SDOF systems undergoing various excitations were tested; the accuracy was found to be excellent when tested against data not seen by the learner, and the computational times were also very low. By predicting over the full domain of the simulated data, time-domain signals were also reconstructed which matched well with the exact solutions. The work shown in this paper indicates the promise of using PINNs for state-estimation in problems where speed of prediction is important. Further work is intended to be done by the authors to apply such a method to more complex systems, as well as to utilise the inherent characteristics of the machine learning approach to produce stochastic information.

REFERENCES

1. Farrar, C. R. and K. Worden. 2007. “An introduction to Structural Health Monitoring,” *The Royal Society – Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 365:303–315.
2. Farrar, C. R. and K. Worden. 2006. “An introduction to structural health monitoring,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sci-*

ences, 365(1851):303–315.

3. Tan, X., W. Chen, T. Zou, J. Yang, and B. Du. 2023. "Real-time prediction of mechanical behaviors of underwater shield tunnel structure using machine learning method based on structural health monitoring data," *Journal of Rock Mechanics and Geotechnical Engineering*, 15(4):886–895.
4. Wittenmark, B., J. Nilsson, and M. Torngren. 1995. "Timing problems in real-time control systems," in *Proceedings of 1995 American Control Conference-ACC'95*, IEEE, vol. 3, pp. 2000–2004.
5. Guo, H., D. Cao, H. Chen, C. Lv, H. Wang, and S. Yang. 2018. "Vehicle dynamic state estimation: State of the art schemes and perspectives," *IEEE/CAA Journal of Automatica Sinica*, 5(2):418–431.
6. Chatzi, E. N. and A. W. Smyth. 2009. "The unscented Kalman filter and particle filter methods for nonlinear structural system identification with non-collocated heterogeneous sensing," *Structural Control and Health Monitoring: The Official Journal of the International Association for Structural Control and Monitoring and of the European Association for the Control of Structures*, 16(1):99–123.
7. Azam, S. E., E. Chatzi, and C. Papadimitriou. 2015. "A dual Kalman filter approach for state estimation via output-only acceleration measurements," *Mechanical systems and signal processing*, 60:866–886.
8. Tatsis, K. E., K. Agathos, V. K. Dertimanis, and E. N. Chatzi. 2022. "A Hierarchical Filtering Approach for Online Damage Detection Using Parametric Reduced-Order Models," in *Special Topics in Structural Dynamics & Experimental Techniques, Volume 5: Proceedings of the 40th IMAC, A Conference and Exposition on Structural Dynamics 2022*, Springer, pp. 103–105.
9. Wang, W.-X., Y.-C. Lai, and C. Grebogi. 2016. "Data based identification and prediction of nonlinear and complex dynamical systems," *Physics Reports*, 644:1–76.
10. Raissi, M., P. Perdikaris, and G. E. Karniadakis. 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.
11. Faroughi, S. A., N. Pawar, C. Fernandes, S. Das, N. K. Kalantari, and S. K. Mahjour. 2022. "Physics-Guided, Physics-Informed, and Physics-Encoded Neural Networks in Scientific Computing," *arXiv preprint arXiv:2211.07377*.
12. Zhong, Y. D., B. Dey, and A. Chakraborty. 2020, "Symplectic ODE-Net: Learning Hamiltonian Dynamics with Control," .
13. Zhang, Z. and C. Sun. 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.
14. Haghighat, E., A. C. Bekar, E. Madenci, and R. Juanes. 2021. "Deep learning for solution and inversion of structural mechanics and vibrations," in *Modeling and Computation in Vibration Problems, Volume 2: Soft computing and uncertainty*, IOP Publishing.
15. Rezaei, S., A. Harandi, A. Moeineddin, B.-X. Xu, and S. Reese. 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.
16. Grossmann, T. G., U. J. Komorowska, J. Latz, and C.-B. Schönlieb. 2023. "Can physics-informed neural networks beat the finite element method?" *arXiv preprint arXiv:2302.04107*.
17. Antonelo, E. A., E. Camponogara, L. O. Seman, E. R. de Souza, J. P. Jordanou, and J. F. Hubner. 2021. "Physics-informed neural nets for control of dynamical systems," *arXiv preprint*

arXiv:2104.02556.

18. Moradi, S., B. Duran, S. Eftekhar Azam, and M. Mofid. 2023. "Novel Physics-Informed Artificial Neural Network Architectures for System and Input Identification of Structural Dynamics PDEs," *Buildings*, 13(3):650.
19. Yuan, F.-G., S. A. Zargar, Q. Chen, and S. Wang. 2020. "Machine learning for structural health monitoring: challenges and opportunities," *Sensors and smart structures technologies for civil, mechanical, and aerospace systems 2020*, 11379:1137903.
20. Sun, L., H. Gao, S. Pan, and J.-X. Wang. 2020. "Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data," *Computer Methods in Applied Mechanics and Engineering*, 361:112732.
21. Wagg, D. and S. Neild. 2010. *Nonlinear vibration with control: for flexible and adaptive structures*, Springer.
22. Sobol, I. 1967. "The distribution of points in a cube and the accurate evaluation of integrals (in Russian) Zh," *Vychisl. Mat. i Mater. Phys*, 7:784–802.