Physics-informed machine learning models for automotive active noise control

M. J. Bryan [1], **O. M. Nielsen** [1,2], **T. Butlin** [1]
[1] University of Cambridge, Department of Engineering
Cambridge, UK

[2] Bose Corporation
Framingham, MA, USA

Abstract

Feedforward Active Noise Control (ANC) can be considered a prediction task: given an input measurement, a model must provide the resulting output both accurately and quickly. This technology is mature in headsets, but reducing structure-borne road noise in vehicles requires the accurate prediction of a strongly nonlinear vibration response in real-time. It can be difficult to develop accurate physics-based models of complex nonlinear systems, whilst purely data-driven approaches generally require impractical amounts of training data. This paper proposes a physics-informed feedback model that separates linear and nonlinear system components, combining a linear physics-based model with a data-driven nonlinear model. The approach is validated on simulated multi-degree-of-freedom systems with varying nonlinearity strengths, and compared with a feedforward model. Results demonstrate improvements in time-domain accuracy and output coherence, especially when considering the number of parameters required.

1 Introduction

1.1 Active Noise Control (ANC)

Active Noise Control (ANC) offers a lightweight and powerful solution for reducing unwanted noise in comparison to passive methods like insulation and damping, and is now ubiquitous in consumer headsets. The principle of ANC is to generate an optimal signal which destructively interferes with the unwanted noise, thereby reducing the sound pressure level (SPL) at a desired location.

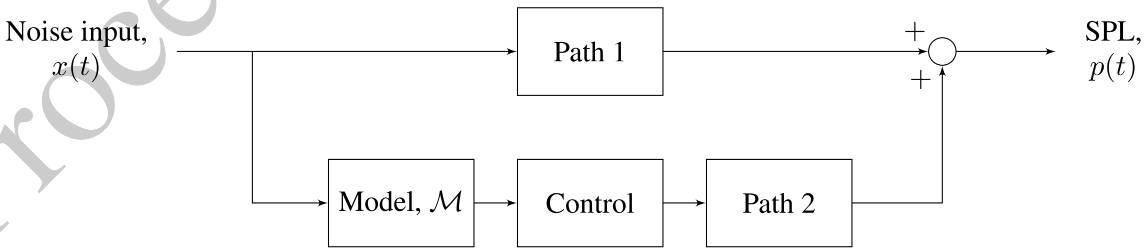

Figure 1: A simplified representation of a feedforward active noise control system.

Figure 1 shows a schematic of a feedforward ANC system, where the noise input $x(t)$ travels through the system (Path 1), resulting in a true output $y_t(t)$, the unmodified sound pressure level (SPL) at the listener's position. The noise input $x(t)$ is measured by a reference sensor, and is fed into a model $\mathcal{M}$ which attempts to provide an optimal reference signal to the controller - the output of which, once filtered by the control path dynamics (Path 2), should destructively interfere with $y_t(t)$ to reduce the SPL at the listener's position, $p(t)$.

In a headset, an external reference microphone is used to measure the incoming noise, and the loudspeaker provides the attenuation signal to reduce the noise in the user's ear. The model $\mathcal{M}$ therefore must be a good model of Path 1 to provide a high quality reference signal, and the performance of the ANC system is highly dependent on its accuracy and latency.

1.2 Challenges in automotive ANC

Feedforward ANC is a mature technology in headsets [1, 2], however reducing structure-borne road noise in vehicles presents additional challenges.

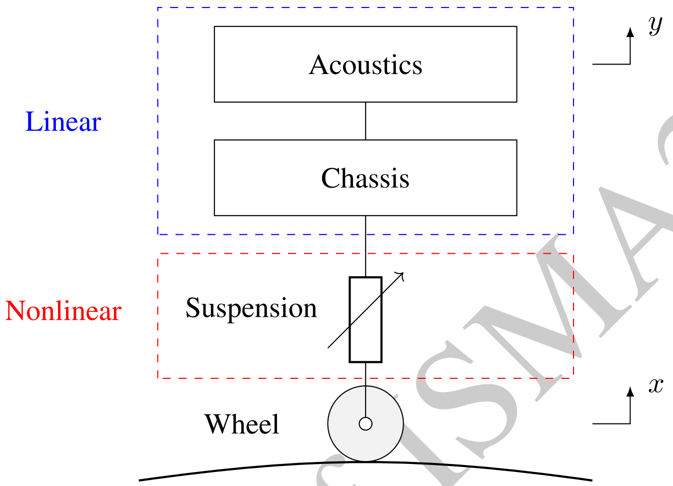

Figure 2: A simplified quarter-car model illustrating the sound transmission pathway.

Figure 2 shows a simplified quarter-car model detailing the key components of the sound transmission pathway in a vehicle. The input displacement $x(t)$ represents the road-induced motion of the wheel hub, which is transmitted through the suspension into the chassis and then the cabin acoustics, resulting in the output SPL $y(t)$. The suspension is coupled to the chassis response, which is coupled to the cabin acoustics.

Previous work by de Brett [3, 4, 5] has shown that the suspension is significantly nonlinear and degrades the coherence between the measured input (wheel hub acceleration) and resulting output (cabin acoustic pressure). Physics-based models for the suspension were also trialled, but showed poor coherence. The chassis and cabin acoustics can be considered effectively linear in comparison with a relatively high modal density, and so their dynamics can be captured with a simpler model.

Measuring across the nonlinearity would mitigate this issue. However, vehicle suspension contains multiple connections into the chassis, each with their own complexities, making it difficult to capture all of the road input. Therefore, wheel hub measurement is required to ensure a high-accuracy reference signal for ANC.

Work by Pike and Cheer [6, 7] has also highlighted the difficulties of ANC in nonlinear systems where the response is input amplitude dependent. They present a technique which uses a set of machine learning (ML) based controllers trained on data from a nonlinear system, before switching between them based on the current operating conditions. This showed improved attenuation performance compared to a classical non-switching control system.

1.3 Physics-informed machine learning

Previous work on ANC in automotive systems has shown that:

1. Purely physics-based models struggle to capture the strong nonlinearity and coupling present [3, 4, 5].
2. Data-driven models need impractical amounts of training data for the necessary performance [8, 9].

Work by Massingham [8, 9] has shown that feedforward convolutional neural networks (CNNs) can be used to model nonlinear systems from an input signal in the context of automotive ANC. However, whilst they are efficient and parallelisable for computational speed, they require an infeasible amount of training data to achieve the necessary performance for ANC.

This motivates the use of physics-informed machine learning (PIML) models to combine the strengths of both approaches. By explicitly encoding the physics of the system in the structure of the model, the aim is to reduce the amount of data required for a given performance, whilst still retaining computational speed and efficiency. PIML is a fast growing area of research yielding impressive results in a wide range of fields from fluid [10, 11] and structural dynamics [12, 13, 14], to materials [15, 16].

1.4 Aims and objectives

This paper focuses on an academic case study of automotive suspension, aiming to:

1. Develop a physics-informed machine learning (PIML) model to provide an improved reference signal for feedforward ANC.

2. Evaluate the performance of the proposed model on a simulated multi-degree-of-freedom (MDOF) system with varying nonlinearity strengths, alongside a suitable feedforward model.

3. Compare these models in the context of ANC, where *both* accuracy and latency are paramount.

2 Idealised models

2.1 Benchmark system

Real automotive systems are highly complex, with multiple degrees of freedom, nonlinearities, and coupling between substructures. Therefore, in order to evaluate the proposed PIML model structure, a highly simplified nonlinear system is used as a benchmark, which captures a minimal set of features of an automotive system: input through a nonlinear element coupled to a linear multi-degree-of-freedom system.

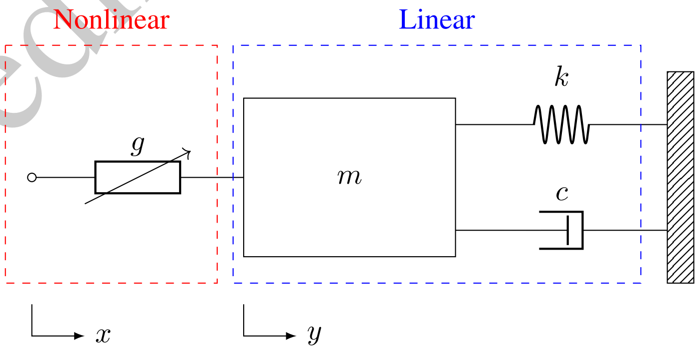

Figure 3: The simplified nonlinear system used as a benchmark.

Figure 3 shows the schematic of the chosen benchmark system, which consists of a multi-degree-of-freedom system, which represents the effectively linear chassis and cabin acoustics, driven through a nonlinear element, which represents the dominant effect of the suspension. The system is driven through a displacement input $x(t)$, and the output is the resulting displacement of the mass $y(t)$ at the driving point. The system is simulated using a second-order Newmark-Beta scheme [17], using a governing equation of the form:

$$m\ddot{y} + c\dot{y} + ky = g(u, \dot{u}) \tag{1}$$

where m, c, and k are the mass, damping, and stiffness of the system, respectively, and $g(u, \dot{u})$ is the nonlinear force input to the system. This is extended to a multi-degree-of-freedom system by exchanging the scalar parameters for matrices, and applying the nonlinear force input to the first degree of freedom only as described in Appendix A.

A cubic or Duffing-type nonlinearity is used as a representative nonlinearity, and is dependent on both the relative displacement and velocity. This is a well-studied form, and mimics the coupled behaviour of a suspension nonlinearity. The relative motion means that the input nonlinear force depends on both the *relative* displacement and velocity, $u = x - y$ and $\dot{u} = \dot{x} - \dot{y}$:

$$g(u, \dot{u}) = k_1 u + k_3 u^3 + c_1 \dot{u} + c_3 \dot{u}^3 \tag{2}$$

where k_1 and c_1 are the linear stiffness and damping coefficients, and k_3 and c_3 are the nonlinear stiffness and damping coefficients respectively. The strength of the nonlinearity can be varied by changing the values of k_3 and c_3, giving a range of systems for model comparison.

The arrangement of the system means that the nonlinearity is coupled to the response of the system, which is a common feature in many mechanical systems. As the output y must be 'fed back' to determine the nonlinear force input, the block diagram representation of the system requires a feedback loop.

Coupling means that the full relationship from the input to the output is highly complex, and so is difficult to model with a feedforward architecture [8]. Nonlinear autoregressive networks with exogenous input or NARX models [18] address this by modelling the system autoregressively using the past inputs and outputs. Such models become extremely compact when they can match the form of the system's equations of motion.

ANC applications necessitate a real-time model, and so a filter-based implementation is desirable. Butlin and Langley [19] developed a model for coupled systems, whereby the linear dynamics of the system are represented by a filter, and the nonlinearity is represented by a static nonlinear function followed by another filter on the return loop. This paper develops the concept by proposing a hybrid physics-ML architecture that can be implemented using filters.

2.2 PIML structure and framework

The feedback representation of coupled nonlinear systems motivates the use of a model structure which explicitly enforces the feedback present. This can be considered a physics-informed machine learning (PIML) model, where the physics information is *encoded* in the structure of the model, and the data is used to learn either or both of the models within it. This feedback model is distinct from a feedback control system.

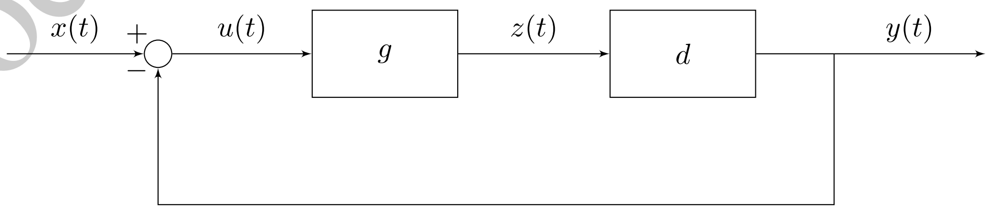

Figure 4: The feedback model expressed as a block diagram.

Figure 4 shows the proposed feedback model structure, where the linear dynamics of the system are represented by impulse response function d, and the nonlinearity is represented by a model g. The input to the nonlinear element g is the relative motion between the input and output, $u = x - y$, meaning that the model's output is fed back into its input. The model g may have memory, so the nonlinear force input to the system $z(t)$ may depend on the current and previous values of u and $\dot{u}$.

The output response in the continuous time domain can be expressed as follows:

$$y(t) = d(t) * g\big(x(t) - y(t)\big) \tag{3}$$

where $d(t)$ is the linear impulse response, $g(\cdot)$ is the nonlinear model, $x(t)$, $y(t)$ are the input and output respectively, and $*$ denotes continuous time convolution.

For the multi-degree-of-freedom (MDOF) benchmark system driven through a nonlinearity in Figure 3, $g(u)$ is the nonlinear function of the relative input u, and the linear dynamics of the system is implemented using the force impulse response of the system:

$$d(t) = \sum_{n=0}^{D} A_n e^{-\zeta_n \omega_n t} \sin(\omega_d t + \phi_n) \tag{4}$$

where A_n, ζ_n, ω_n, and ϕ_n are the chosen parameters (amplitude, natural frequency, damping ratio, and phase) for mode n, and the damped natural frequency is given by $\omega_d = \omega_n \sqrt{1 - \zeta_n^2}$.

Summing over the D modes, discretising with respect to the time step Δ, and truncating to a length of L_d timesteps (chosen to respect the system's time constant) gives the following discrete-time convolution for a timestep k, where z is the nonlinear force input:

$$d[k] = \sum_{n=0}^{D} A_n e^{-\zeta_n \omega_n k\Delta} \sin(\omega_d k\Delta + \phi_n) \tag{5}$$

$$y[k] = \sum_{i=k-L_d}^{k} d[k-i] \cdot z[k] \tag{6}$$

In a practical system, the linear dynamics d could be obtained from a physics-based model, such as a finite element model, or from experimental modal analysis, whilst the nonlinearity could be learned from data using a data-driven model, such as a neural network or dictionary method.

The key contribution of the feedback loop is the ability to exploit knowledge of the form of the equations of motion, targeting a more efficient learning of the nonlinearity, focusing model power on g.

The physics of the nonlinearity are generally unknown, but typically have a short memory [9]. On the other hand, the overall system memory is long due to the long time constants of linear dynamics d and the recursively coupled system. Therefore, the nonlinearity g in isolation usually has a shorter overall memory than the full system, and so can be represented by a smaller ML model.

There are many choices for the model g, and a CNN is chosen as a powerful and efficient architecture for modelling complex functions with short-term dependencies. An recursive neural network (RNN) could also be employed, however, CNNs are more efficient to train and evaluate as they can be parallelised in time, whereas RNNs must be evaluated auto-regressively.

This hybrid approach should be more efficient than a fully feedforward framework which directly models the system from the input - a feedforward CNN will serve as the comparative benchmark (Section 2.4).

2.3 Training and prediction modes

A real time model must be able to operate autoregressively i.e. predict the current output using only knowledge of the past inputs and predicted outputs:

$$y_p[k] = \mathcal{M}\Big(x[0:k], y_p[0:k-1]\Big) \tag{7}$$

As the model d is the linear force-displacement impulse response of the system, its first element is zero for systems with non-zero mass ($d[0] = 0$). This prevents an *algebraic loop* as the input at time step k has no influence on the output at time step k, and only affects subsequent timesteps.

However, training a model autoregressively is computationally intensive [18, 20] as it cannot be mass-parallelised, and can result in stability issues. Therefore, a more efficient training method is used, which trains the model 'one step ahead' using the true output y_t at the previous time steps as input, rather than the predicted output y_p. Because the model input is always equal to $u = x - y$, using the true output y_t during training does not result in target leakage whereby the model would learn to simply copy the output.

This method is highly efficient and parallelisable, allowing the computation of the whole time series for the whole batch in one forward pass. Therefore, this 'one step ahead' mode is used for training, and the autoregressive mode in Equation 7 is used for prediction.

2.4 Comparison with feedforward models

A CNN is used as the feedforward benchmark for comparison with the proposed feedback model. This is intended to be an example of a high-performance, efficient feedforward architecture, rather than an attempt to find the 'best' model.

Standard CNNs have been shown to be powerful for modelling nonlinear systems [21], and the WaveNet architecture [22] is an extension that uses exponential dilation [23]. This allows the efficient modelling of long-term time dependencies, and has previously shown excellent performance on nonlinear systems [9].

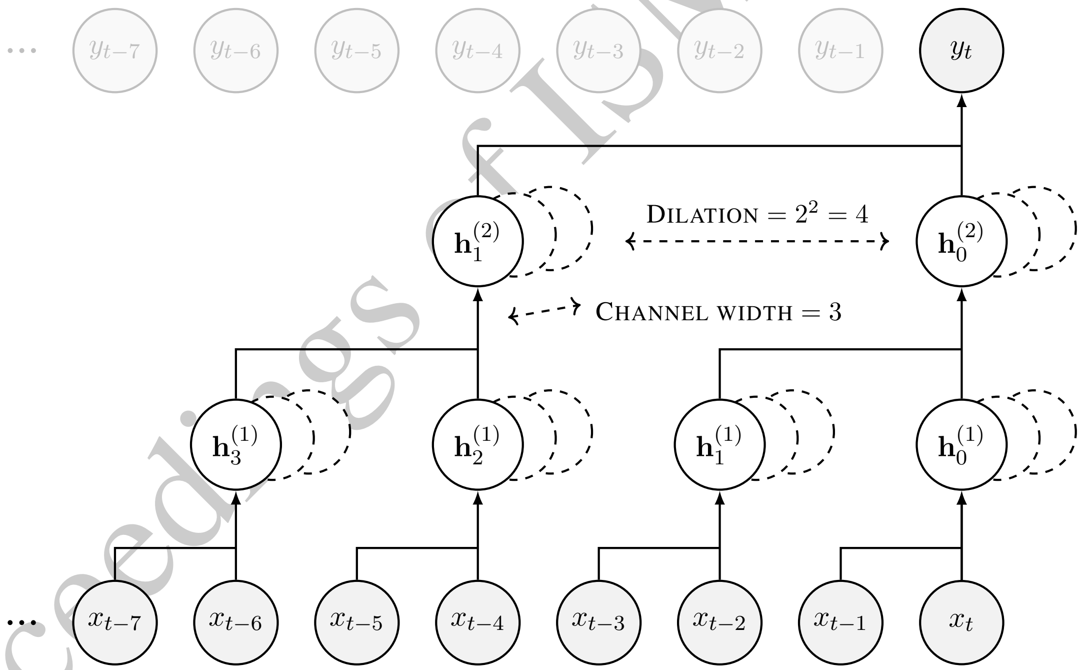

Figure 5: Diagram representation of a WaveNet CNN with two hidden layers, a kernel width of two, and dilation between nodes of 2^l, where l is the l^{th} hidden layer, adapted from Massingham [9].

Figure 5 shows an example WaveNet CNN (WNCNN) architecture where the output of a convolution layer $\mathbf{h}^{(l)}$ is given by:

$$\mathbf{h}^{(l)} = \mathcal{A}\Big(\mathbf{W}^{(l)} * \mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\Big) \tag{8}$$

where $\mathbf{W}^{(l)}$ is the kernel of convolution layer l, $\mathbf{h}^{(l-1)}$ is the previous layer output, $\mathbf{b}^{(l)}$ is the bias, and $\mathcal{A}$ is a nonlinear activation function, in this case a ReLU: $\mathcal{A}(x) = \max(0, x)$.

The aim is to evaluate the performance of the proposed feedback model against a similarly parameter-efficient feedforward model in order to determine whether the feedback structure intrinsically provides a benefit in terms of accuracy or efficiency.

3 Methodology

3.1 Dataset generation

The system described in Section 2.1 is simulated using a second-order Newmark-Beta scheme [17] subject to broadband input noise, which has a white velocity spectrum to broadly mimic road roughness [24].

An ensemble of independent frames is generated, which allows for robust training without the need to window for subsequent calculation of frequency-domain quantities. A single simulation of the system is carried out, before splitting the input x and output y_t into a dataset of N frames, each of length L time steps.

3.2 Model training and evaluation

3.2.1 Training and prediction

Both the proposed feedback model and the feedforward WNCNN are implemented in PyTorch [25] using Conv1D layers, and trained using the Adam optimiser [26] with a learning rate of 10^{-3} for 1000 epochs. The training dataset contains $N = 2000$ frames of length $L = 2000$ time steps, and the test dataset contains $N = 500$ frames of the same length.

When training, the feedback model utilises the 'one step ahead' method described in Section 2.3 for computational efficiency, and the autoregressive method in Equation 7 is used for prediction to mimic practical use.

3.2.2 Loss criterion

The loss criterion used throughout is the normalised mean-squared error (NMSE) between the true output $y_t(t)$ and the predicted output $y_p(t)$ of the model, defined as:

$$\mathcal{L}_{\text{NMSE}}(y_t, y_p) = \mathbb{E}\left[\left(\frac{y_t(t)}{\sigma_y} - \frac{y_p(t)}{\sigma_y}\right)^2\right] \tag{9}$$

$$= \frac{1}{\sigma_y^2 NL} \sum_{i=1}^{N} \sum_{k=1}^{L} \left(y_t^{(i)}[k] - y_p^{(i)}[k]\right)^2 \tag{10}$$

where N is the number of frames in the dataset and L is the length of each frame in time steps. Both terms are normalised by the standard deviation of the true output σ_y for better comparison across different systems and models.

3.2.3 Output coherence

Whilst the models described are trained on the time-domain normalised mean-squared error loss described in Equation 10, the performance of models for ANC is commonly evaluated using frequency-domain metrics such as coherence. Coherence provides a measure for how much of the output can be explained by the input via a linear relationship, and takes a value between 0 and 1, where 1 indicates perfect correlation.

In nonlinear systems, linear coherence is always reduced, degrading ANC performance. Therefore, the 'output coherence' of a model [9] can be defined as the linear coherence between true output $y_t(t)$ and its predicted output $y_p(t)$:

$$\gamma^2_{y_t y_p}(\omega) = \frac{|S_{y_t y_p}(\omega)|^2}{S_{y_t y_t}(\omega) S_{y_p y_p}(\omega)} \tag{11}$$

where y_t is the true output of the system, y_p is the predicted output of the model, ω is the frequency, and $S_{ab}(\omega)$ is the power spectral density between signals a and b.

A good model should improve this coherence over the linear coherence between the input and output, thereby capturing more of the nonlinearity present. This makes output coherence a valuable metric to evaluate the performance of the proposed feedback model, and compare to it with a standard feedforward model in the context of ANC.

4 Results

4.1 Experimental outline

Two experiments were carried out to evaluate the performance of the proposed feedback model and compare it with a feedforward WNCNN. The first experiment evaluates the performance of the two models as the strength of the nonlinearity increases, and the second compares their performance across different model sizes to evaluate their parameter efficiencies.

Table 1: System parameters for the two benchmark nonlinear systems.

Parameter	**Symbol**	**Unit**	**System 1**	**System 2**
Natural frequencies	ω_n	Hz	[2, 5, 7]	same
Damping ratios	ζ_n	–	[0.1, 0.1, 0.1]	same
Linear stiffness	k_1	$\mathrm{N\,m^{-1}}$	1	1
Nonlinear stiffness (displacement)	k_3	$\mathrm{N\,m^{-3}}$	[0, 10, 1000]	0
Linear damping	c_1	$\mathrm{N\,s\,m^{-1}}$	0	1
Nonlinear damping (velocity)	c_3	$\mathrm{N\,s^3\,m^{-3}}$	0	[0, 0.1, 1]
Time step	Δ	s	10^{-2}	same
Linear time constant	τ_n	time steps	[80, 32, 23]	same
Impulse response length	L_d	time steps	400	same

For this, two nonlinear benchmark systems as described in Section 2.1 were used with their parameters detailed in Table 1. The first using a displacement nonlinearity of varying strength (System 1), and the second using a velocity nonlinearity of varying strength (System 2), both driving the same underlying three-degree-of-freedom linear system.

Also given in Table 1 is the simulation time step Δ, linear time constant $\tau_n = 1/\zeta_n\omega_n$, and impulse response length used for model d - chosen to be five times the time constant of the slowest mode to ensure that the linear dynamics are captured sufficiently.

Both model types are trained on the same datasets for each system ($N = 2000$ frames of training data, $N = 500$ frames of test data, each of length $L = 2000$ time steps), and the performance is compared in terms of their best fit linear transfer function, output coherence, and final test loss.

For the first experiment in Section 4.2.2, the same model architectures (number of layers, kernel width, activation function) are used for each strength of nonlinearity, and the performance is compared across the strengths. In Section 4.2.3, the number of parameters in each model is varied by changing the CNN channel width as shown in Figure 5, with a wider channel width allowing the model more power. Performance is then compared across the different numbers of parameters for each system.

4.2 Performance

4.2.1 Example output

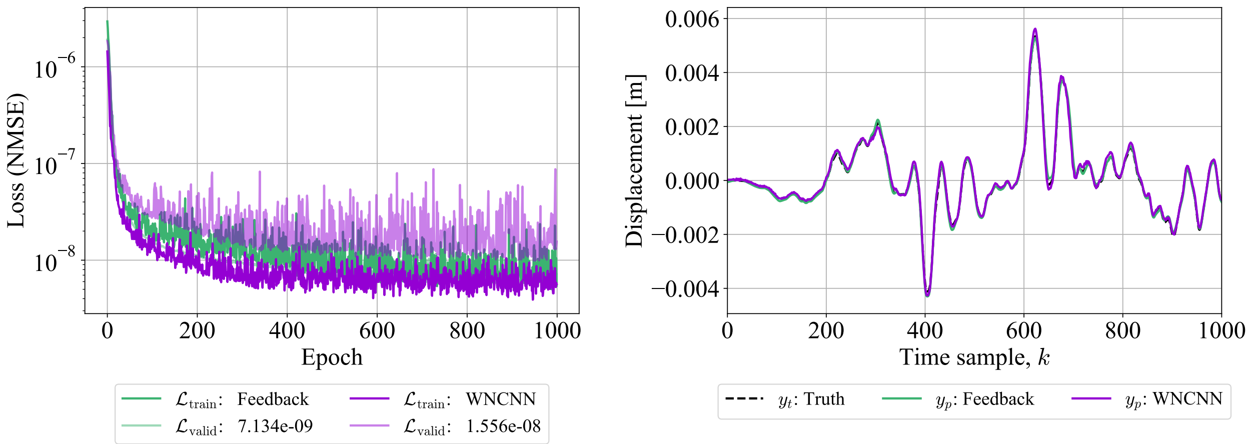

Figure 6: Example loss curves (left) and time series prediction (right) for a feedback model and WNCNN on System 1 with a weak displacement nonlinearity of $k_3 = 10$.

Figure 6 shows an example of the training loss curves and time series prediction for a feedback model and WNCNN on System 1 with a weak displacement nonlinearity of $k_3 = 10$. Both models show similar training behaviour, however, the feedback model achieves better generalisation, with a more stable validation loss with a lower final value (see legend). Time series prediction for both models show good agreement with the true output, and so transfer function and output coherence measures must be used to discern any difference.

4.2.2 Strength of nonlinearity

For each system with varying nonlinearity, the best fit linear transfer function, output coherence, and final test loss are computed on the test dataset for the feedback model and feedforward WNCNN for comparison.

The best fit linear transfer function, often referred to as the G_1 estimator or first Wiener kernel W_1, is computed directly using the cross-spectral density between the input and predicted output, and the power spectral density of the input:

$$W_1(\omega) = \frac{S_{xy_p}(\omega)}{S_{xx}(\omega)} \tag{12}$$

where $S_{xy_p}(\omega)$ is the cross-spectral density between the input $x(t)$ and predicted output $y_p(t)$, and $S_{xx}(\omega)$ is the power spectral density of the input. This is a complex quantity, but the magnitude is used to assess how well the model captures the linearised dynamics of the system in the frequency domain.

The output coherence is computed as described in Section 3.2.3, and is compared to the input or linear coherence of the system ($\gamma^2_{xy}(\omega)$: black, dashed) to evaluate how the models improve the system coherence. The final test loss is computed using NMSE as in Equation 10, and is shown as a bar for comparison.

Figure 7 shows the linearised transfer function, output coherence, and final test loss for System 1 with varying displacement nonlinearity. Across the three strengths of nonlinearity, both the feedback and WCNN capture W_1 accurately even as the nonlinearity increases. Both models also show a coherence improvement over the linear coherence, but the feedback model shows a greater improvement as the nonlinearity increases, especially at higher frequencies. However, both models still show a coherence reduction at frequencies approaching the Nyquist frequency of the simulation (50 Hz).

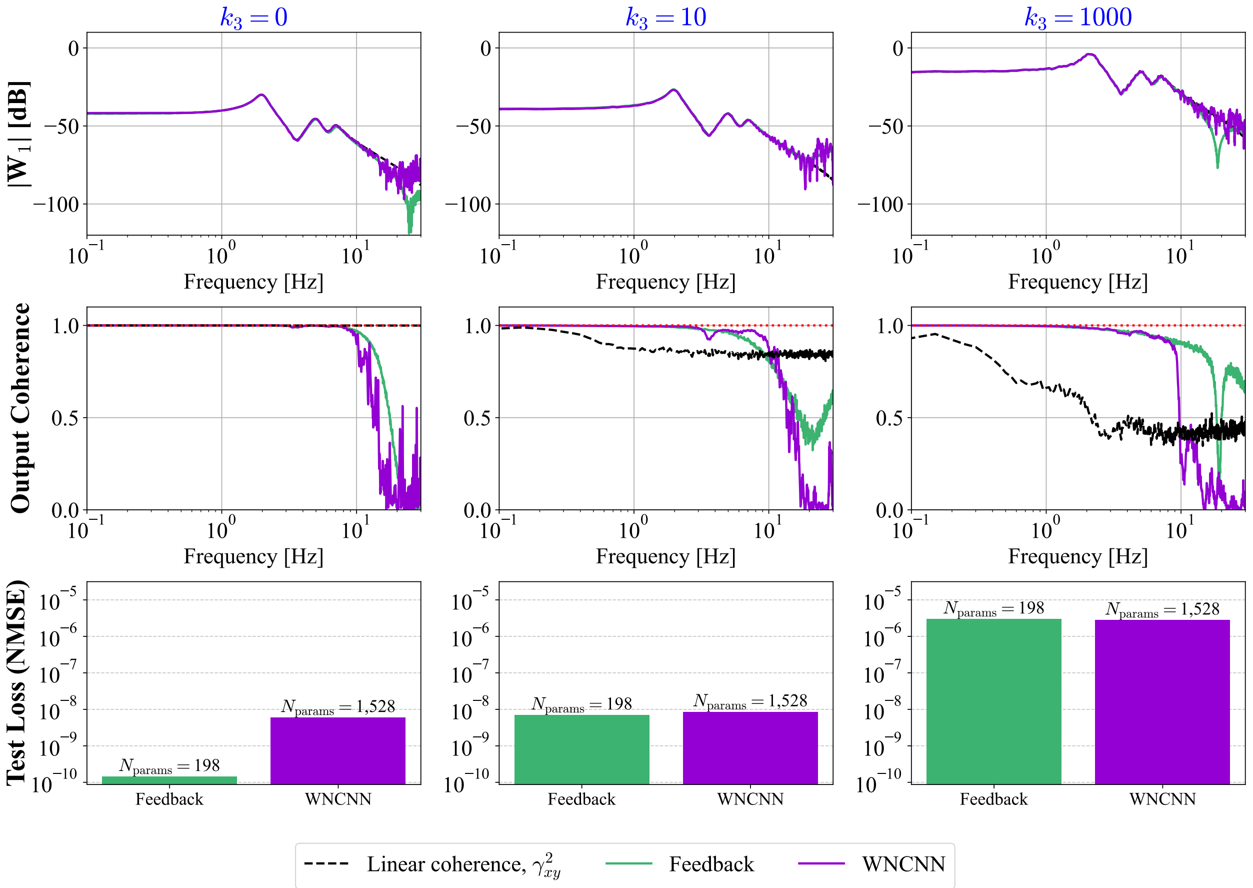

Figure 7: First Wiener kernel W_1, output coherence, and final test loss for System 1, with columns for increasing displacement nonlinearity strength, $k_3 = [0, 10, 1000]$.

Test loss increases with nonlinearity strength for both models. The feedback model achieves better accuracy in the linear case and comparable accuracy in the nonlinear cases with significantly fewer parameters, demonstrating the efficiency of the feedback structure.

Figure 8 details similar results for a velocity nonlinearity (System 2). As with System 1, both models capture W_1 accurately across the strengths of nonlinearity, but deviate closer to the Nyquist frequency. Linear coherence in all three cases is already relatively high, however, both models improve this in the nonlinear cases ($c_3 = 0.1$ and $c_3 = 1$), with the feedback model doing so across a slightly wider frequency range compared to the WNCNN.

For the most nonlinear case ($c_3 = 1$), a deeper CNN for g in the feedback model with 643 rather than 198 parameters was required to capture the behaviour, making it three times larger than the previous examples, but still more accurate and significantly smaller than the WNCNN.

Again, the final test loss increases with the strength of nonlinearity for both models. But for the velocity nonlinearity, the feedback model shows an improvement over the WNCNN for all cases. This nonlinearity has memory, unlike the displacement nonlinearity, as the model g must capture the velocity $\dot{u}$ of the system from the relative displacement u, where a simple, first-order accurate estimate is the backward Euler method:

$$\dot{u}[k] \approx \frac{u[k] - u[k-1]}{\Delta} \tag{13}$$

This may explain the greater improvement, as the feedback structure allows the modelling of the nonlinearity and its short memory separately, rather than the full, longer system memory with a feedforward model.

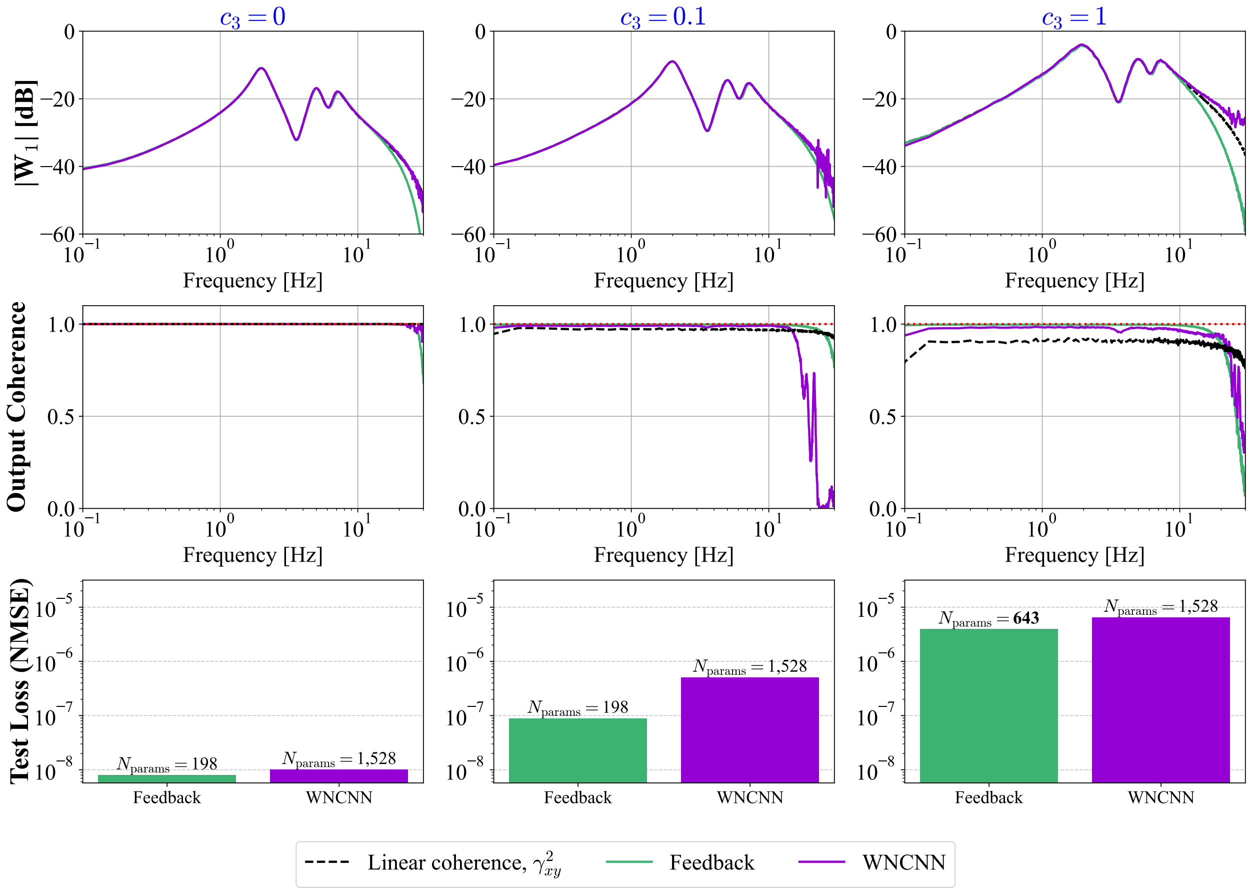

Figure 8: First Wiener kernel W_1, output coherence, and final test loss for System 2, with columns for increasing velocity nonlinearity strength, $c_3 = [0, 0.1, 1]$.

4.2.3 Number of parameters

To explore the parameter efficiency benefit highlighted in Section 4.2.2, the same models are trained with varying numbers of parameters to assess how each model type scales. This is achieved by changing the channel width as described in Section 4.1, ranging from 10^2 to 10^4 parameters. The same training and evaluation procedure is used as before, and the final test loss is plotted against the number of parameters for each model type and strength of nonlinearity.

For each model type and size, five independent models are trained on different random initialisations, then evaluated on the same test dataset, aiming to highlight the stochastic nature of model training, which can lead to outliers: 'failed' or poor performaing models that get stuck in local loss minima during training.

Scaling laws have been observed in the behaviour of many data-driven models [27] and have been applied to models for dynamical systems [9], showing power law behaviour. Therefore a line of best fit (LoBF) is fitted to the points for each model type and strength of nonlinearity to highlight the general loss-parameter trend, rather than aim to provide the exact scaling law for the model.

Figure 9 shows scaling for the displacement nonlinearity system (System 1). Test loss increases with nonlinearity strength, and decreases with model size as expected. The feedback model outperforms the WNCNN for each strength at the same size. This is especially clear in the linear case, whereby the feedback model requires fewer than 100 parameters to achieve a low test loss, and does not improve significantly as the model size increases. The WCNN instead requires many more parameters to achieve the same performance.

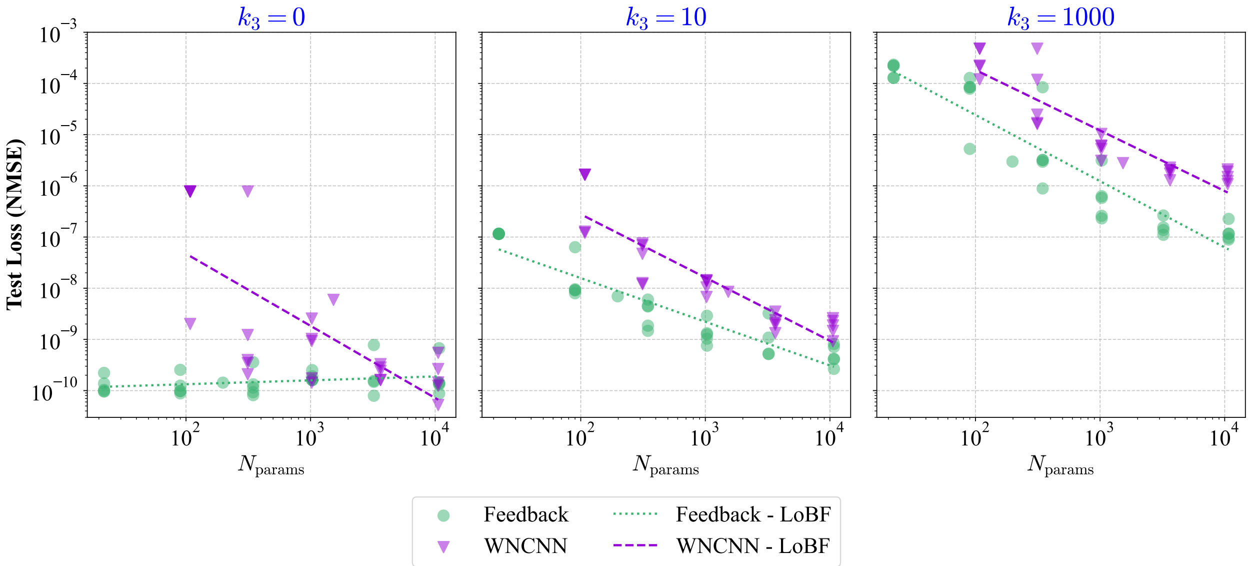

Figure 9: Test loss versus numbers of parameters, N_{params}, for the feedback and WNCNN models on the displacement nonlinearity system (System 1).

For the nonlinear cases ($k_3 = 10$ and $k_3 = 1000$), the feedback model also outperforms the WNCNN across all model sizes, highlighted by a similar trend offset by approximately half a decade showing that the feedback structure is beneficial for parameter efficiency.

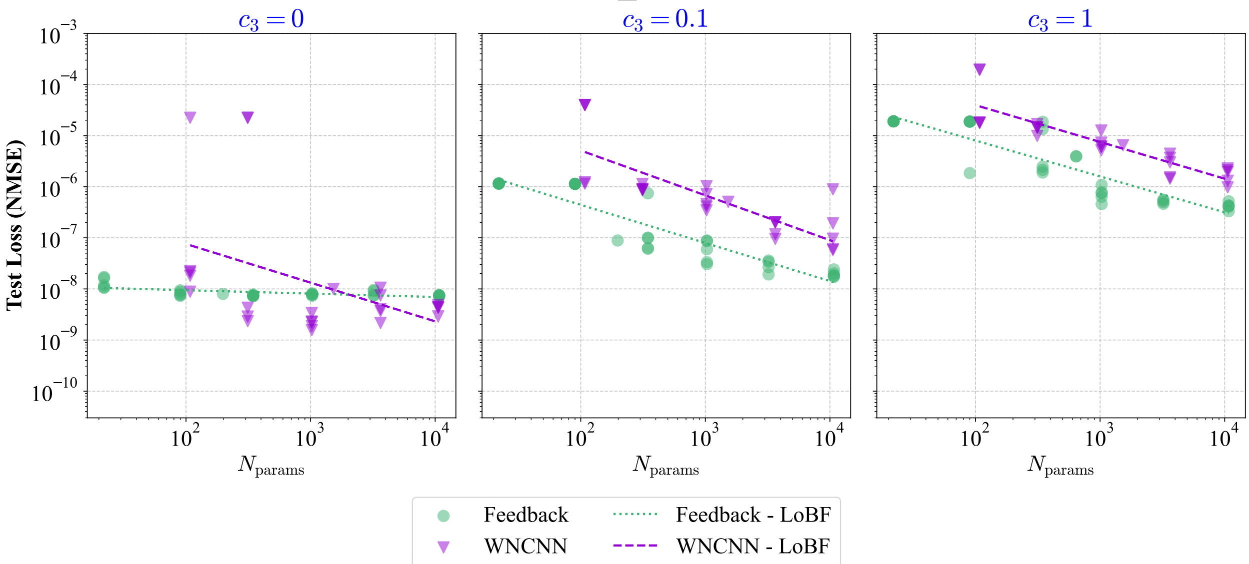

Figure 10: Test loss versus numbers of parameters, N_{params}, for the feedback and WNCNN models on the velocity nonlinearity system (System 2).

The general trends are the same for the velocity nonlinearity system in Figure 10. For the linear case ($c_3 = 0$), the feedback model requires few parameters and shows little improvement, however, the WNCNN is an better model when allowed more parameters in this simple case.

For the nonlinear cases ($c_3 = 0.1$ and $c_3 = 1$), the feedback model once again outperforms the WNCNN across all model sizes with a similar trend offset by approximately half a decade, showing the efficiency of the feedback structure. This suggests that the efficiency benefit of the feedback structure becomes clearer as the strength of nonlinearity increases, for both displacement and velocity dependent functions.

5 Conclusions

Modelling nonlinear systems like automotive suspension remains a difficult task, even with machine learning techniques. This work has shown that by enforcing a feedback structure within a model to separate the linear and nonlinear components of the system, model efficiency can be improved with equal or often greater accuracy than a feedforward model, especially as the strength of the nonlinearity increases. This is a promising result for the application of physics-informed machine learning (PIML) to nonlinear systems, and motivates further work on more complex systems, and with more representative benchmarks.

In the future, the proposed feedback model will be tested on a more representative practical benchmark: a coupled plate system driven through a nonlinearity, which provides an additional challenge in representing the weaker coupling between the chassis and acoustics, as well as real measurement noise. Additionally, the model will be extended to a framework whereby the linear and nonlinear models can be either known, trained, or a combination of the two, allowing for a more flexible framework which can be adapted to different systems and applications.

Acknowledgements

This work was funded by Bose Corporation, whom the authors would like to thank for their support, alongside Dr Joe Massingham for helpful technical discussions.

References

[1] A. G. Bose and J. Carter, "Headphoning," U.S. Patent US4 455 675A, Jun., 1984.

[2] N. V. George and G. Panda, "Advances in active noise control: A survey, with emphasis on recent nonlinear techniques," *Signal Processing*, vol. 93, no. 2, pp. 363–377, Feb. 2013.

[3] M. De Brett, "Implications of nonlinear suspension behaviour for feedforward control of road noise in cars," Ph.D. dissertation, University of Cambridge, Apr. 2021.

[4] M. De Brett, T. Butlin, L. Andrade, and O. M. Nielsen, "Experimental investigation into the role of nonlinear suspension behaviour in limiting feedforward road noise cancellation," *Journal of Sound and Vibration*, vol. 516, p. 116532, Jan. 2022.

[5] M. De Brett, T. Butlin, and O. M. Nielsen, "Analysis of nonlinear vibration transmission through a vehicle suspension damper at low audio frequencies," *Journal of Sound and Vibration*, vol. 551, p. 117615, May 2023.

[6] X. Pike and J. Cheer, "Comparison of neural network architectures for feedforward active control of nonlinear systems," in *Proceedings of ISMA24*, 2024.

[7] X. Pike and J. Cheer, "Dynamic neural network switching for active control of nonlinear systems," *The Journal of the Acoustical Society of America*, vol. 158, no. 1, pp. 154–163, Jul. 2025.

[8] J. Massingham, O. Nielsen, and T. Butlin, "Convolutional neural networks applied to nonlinear dynamical systems: a Wiener series perspective," in *Proceedings of ISMA2024*, Leuven, Belgium, 2024.

[9] J. Massingham, "Performance Limits of Active Noise Control for Nonlinear Systems: A Machine Learning Perspective," PhD Thesis, University of Cambridge, Nov. 2025.

[10] Y. Sun, U. Sengupta, and M. Juniper, "Physics-informed deep learning for simultaneous surrogate modeling and PDE-constrained optimization of an airfoil geometry," *Computer Methods in Applied Mechanics and Engineering*, vol. 411, p. 116042, Jun. 2023.

[11] M. Tomasetto, J. P. Williams, F. Braghin, A. Manzoni, and J. N. Kutz, "Reduced order modeling with shallow recurrent decoder networks," *Nature Communications*, vol. 16, no. 1, p. 10260, Nov. 2025.

[12] Z. Lai, C. Mylonas, S. Nagarajaiah, and E. Chatzi, "Structural identification with physics-informed neural-odes," *Journal of Sound and Vibration*, vol. 508, p. 116196, Sep. 2021.

[13] M. Haywood-Alexander, W. Liu, K. Bacsa, Z. Lai, and E. Chatzi, "Discussing the spectrum of physics-enhanced machine learning: a survey on structural mechanics applications," *Data-Centric Engineering*, vol. 5, p. e31, Jan. 2024.

[14] D. Pitchforth, M. Jones, S. Gibson, and E. Cross, "Physics-informed kernel mixtures for structural dynamics," *Mechanical Systems and Signal Processing*, vol. 254, p. 114330, Jun. 2026.

[15] M. Maia, I. Rocha, P. Kerfriden, and F. Van Der Meer, "Physically recurrent neural networks for path-dependent heterogeneous materials: Embedding constitutive models in a data-driven surrogate," *Computer Methods in Applied Mechanics and Engineering*, vol. 407, p. 115934, Mar. 2023.

[16] Z. Li, H. Zheng, N. Kovachki, D. Jin, H. Chen, B. Liu, K. Azizzadenesheli, and A. Anandkumar, "Physics-Informed Neural Operator for Learning Partial Differential Equations," *ACM / IMS J. Data Sci.*, vol. 1, no. 3, pp. 9:1–9:27, May 2024.

[17] N. M. Newmark, "A Method of Computation for Structural Dynamics," *Journal of the Engineering Mechanics Division*, vol. 85, no. 3, pp. 67–94, Jul. 1959.

[18] S. A. Billings, *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*. Wiley, 2013.

[19] T. Butlin and R. Langley, "An efficient model of drillstring dynamics," *Journal of Sound and Vibration*, vol. 356, pp. 100–123, Nov. 2015.

[20] K. Worden and G. R. Tomlinson, *Nonlinearity in Structural Dynamics*. IOP Publishing, 2001.

[21] M. Lopez and W. Yu, "Nonlinear system modeling using convolutional neural networks," in *2017 14th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, Oct. 2017, pp. 1–5.

[22] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," Sep. 2016, arXiv:1609.03499.

[23] F. Yu and V. Koltun, "Multi-Scale Context Aggregation by Dilated Convolutions," Apr. 2016, arXiv:1511.07122.

[24] P. Andrén, "Power spectral density approximations of longitudinal road profiles," *International Journal of Vehicle Design - INT J VEH DES*, vol. 40, Jan. 2006.

[25] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," Dec. 2019, arXiv:1912.01703.

[26] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Jan. 2017, arXiv:1412.6980.

[27] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling Laws for Neural Language Models," Jan. 2020.

[28] M. E. McIntyre and J. Woodhouse, "On the Fundamentals of Bowed-String Dynamics," *Acustica*, vol. 43, no. 2, pp. 93 – 108, 1979.

Appendix

A Newmark-Beta scheme

The Newmark-Beta method [17] is second-order numerical integration scheme, here used to solve discretised ordinary differential equations of the form:

$$\mathbf{M}\ddot{\mathbf{y}} + \mathbf{C}\dot{\mathbf{y}} + \mathbf{K}\mathbf{y} = \mathbf{f}_{\text{nl}} \tag{14}$$

where $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{K}$ are the mass, damping, and stiffness matrices of the system, $\mathbf{y}$ is the output vector, and $\mathbf{f}_{\text{nl}}$ is the input nonlinear force vector.

For a system with D degrees of freedom, if the input is applied to and the output measured at the same degree of freedom y, the system can be reduced using driving point model by using model coordinates $\mathbf{q}$ and a modal transformation matrix $\mathbf{\Phi}$, such that:

$$\mathbf{y} = \mathbf{\Phi}\mathbf{q} \tag{15}$$

$$\begin{pmatrix} y \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_{D-1} \end{pmatrix} \tag{16}$$

Combining with Equation 14 gives a governing equation for the modal coordinates $\mathbf{q}$:

$$\mathbf{M}'\ddot{\mathbf{q}} + \mathbf{C}'\dot{\mathbf{q}} + \mathbf{K}'\mathbf{q} = \mathbf{\Phi}^T\mathbf{f}_{\text{nl}} \tag{17}$$

where $\mathbf{M}' = \mathbf{\Phi}^T\mathbf{M}\mathbf{\Phi}$ for each of the matrices. This can be expressed in terms of ω_n, the natural frequencies in $\mathrm{rad\,s^{-1}}$, and ζ_n, the corresponding damping ratios, which are the specified modal parameters of the system:

$$\mathbf{I}\ddot{\mathbf{q}} + \text{diag}(2\zeta_1\omega_1, \dots)\dot{\mathbf{q}} + \text{diag}(\omega_1^2, \dots)\mathbf{q} = \mathbf{\Phi}^T\left[f_{\text{nl}}, 0, \dots, 0\right]^T \tag{18}$$

where $\mathbf{I}$ is the $D \times D$ identity matrix, and $\text{diag}(\cdot)$ is the diagonal operator across all D modes.

The method is then iterative, using a Newton-Raphson scheme to converge the nonlinear force $\mathbf{f}_{\text{nl}}$ at each time step (described by McIntyre and Woodhouse [28]). This nonlinear force combined with the internal force from the system gives an implicit equation for the acceleration at each time step, which can be solved using the Newmark-Beta method:

$$\mathbf{q}_{k+1} = \mathbf{A}'^{-1}\Big(\mathbf{\Phi}^T\mathbf{f}_{\text{int},k+1} + \mathbf{f}_{\text{nl},k}\Big) \tag{19}$$

$$\dot{\mathbf{q}}_{k+1} = \frac{\gamma}{\beta\Delta}(\mathbf{q}_{k+1} - \mathbf{q}_k) + \Big(1 - \frac{\gamma}{\beta}\Big)\dot{\mathbf{q}}_k + \Delta\Big(1 - \frac{\gamma}{2\beta}\Big)\ddot{\mathbf{q}}_k \tag{20}$$

$$\ddot{\mathbf{q}}_{k+1} = \frac{1}{\beta\Delta^2}(\mathbf{q}_{k+1} - \mathbf{q}_k) - \frac{1}{\beta\Delta}\dot{\mathbf{q}}_k - \Big(\frac{1}{2\beta} - 1\Big)\ddot{\mathbf{q}}_k \tag{21}$$

where $\mathbf{A}' = \frac{1}{\beta\Delta^2}\mathbf{M}' + \frac{\gamma}{\beta\Delta}\mathbf{C}' + \mathbf{K}'$ is the effective stiffness matrix, $\mathbf{f}_{\text{int},k+1}$ is the internal force at time step $k+1$, and $\mathbf{f}_{\text{nl},k}$ is the nonlinear force at time step k. The parameters β and γ are chosen to ensure stability and accuracy of the method without numerical damping ($\beta = 0.25$ and $\gamma = 0.5$).

This allows simulation of a multi-degree-of-freedom system driven through a nonlinearity, whereby the force is a function of the relative displacement and velocity of the system as described in Section 2.1.