forum acusticum 2023

GENERALIZED PERFORMANCE OF NEURAL NETWORK CONTROLLERS FOR FEEDFORWARD ACTIVE CONTROL OF NONLINEAR SYSTEMS

Xander Pike[1*] **Jordan Cheer**[1]
[1] ISVR, University of Southampton, United Kingdom

ABSTRACT

Over the past few decades, advances in digital technologies have allowed for the development of complex active control solutions for both vibration and acoustic control and have been utilised in a wide range of applications. Such control systems are commonly designed using linear filters which cannot fully capture the dynamics of nonlinear systems. To overcome such issues, it has previously been shown that replacing linear controllers with Neural Networks (NNs) can improve control performance in the presence of nonlinearities in both the system plant and primary path. However, the performance of the controller across excitation levels has not been frequently explored. Controllers with good performance across a range of excitation levels would be essential in the control of many real systems. In this paper, a method of training Multilayer Perceptrons (MLPs) for single-input-single-output (SISO) feedforward acoustic noise control is presented. In a simple time-discrete simulation, the performance of the trained NNs is investigated for different excitation levels. The effects of the properties of the training data and NN controller on generalized performance are explored.

Keywords: *machine learning, active control, neural network, nonlinear*

**Corresponding author*: X.Pike@soton.ac.uk.

1. INTRODUCTION

Unwanted noise and vibration can be problematic in both engineering systems and in public and private spaces. Passive control solutions are capable of effectively eliminating high frequency components of noise and vibration but are typically large or heavy, possibly exceeding the design constraints of a given application. Active control solutions, by contrast, are capable of good control at low frequencies, and are typically lightweight and small in size. Historically, feedforward active noise and vibration control systems have been implemented using linear control filters and linear plant models, commonly using the famous FxLMS algorithm. However, it is well understood that nonlinearities present in either the plant or primary path of the control system can have a significant impact on control performance [1–4]. Many approaches have been proposed to overcome this limitation, including polynomial, cross-term or trigonometric expansion of the reference signal [5, 6], genetic algorithms [7] and fuzzy logic-based methods [8]. A further common approach, which has been applied to active control over the past few decades, is the use of machine learning. NNs in particular are known to possess the property of being 'universal approximators' [9] and are therefore an attractive black-box method for the modelling and control of unknown or uncertain nonlinear systems. Many different uses of NNs have been studied, including plant/system modelling [4, 10–13], feedforward controller design [4, 10, 14], inverse modelling [15], signal prediction and feedback control [16–20], linear filter selection [21], adaptive parameter estimation for linear controllers [20, 22], frequency-domain control [23], multichannel controller design [24], and signal classification [25]. The similarity in structure between NNs and linear filters provides good motivation

for their use in both system modelling and feedforward controller design. Inpevious work utilising NNs as feedforward controllers, however, it is uncommon to see an exploration of the ability of the neural networks to generalize across a range of excitation levels of the system in question. This is clearly a desirable quality in any real implementation of such a control system where the properties of the excitation, and therefore the effect of the system nonlinearity, may change over time. In this paper, a time-discrete simulation of a simple acoustic noise control system using an MLP is studied. Section 2 defines the simulation setup, system parameters and simulation method. Section 3 explains the controller training methodology. Section 4 presents simulated results in the time and frequency domains. Finally, Section 5 discusses conclusions and presents possible future research directions.

2. PROBLEM DEFINITION

2.1 Simulated system

The considered system consists of two acoustic sources. The primary source, which generates the acoustic disturbance, is modelled as a damped Duffing oscillator, which radiates as a monopole acoustic source. The secondary acoustic source, which generates the cancelling acoustic signal, is modelled as a simple harmonic oscillator, which also radiates as a monopole source. Figure 1 shows a diagram of the assumed arrangement.

The displacement of the Duffing oscillator, $y_a(t)$, is caused by the motion of the floor to which it is attached. The displacement $x(t)$ of this floor is also taken to be the reference signal passed to the feedforward controller. The displacement $y_b(t)$ of the mass m_b is caused by the control force $F_c(t)$ produced by the controller.

The equations of motion for the total system are

$$m_a\ddot{y}_a(t) + k_a p(t) + k_a^{NL} p^3(t) + c_a\dot{p}(t) = 0 \quad (1)$$

$$m_b\ddot{y}_b(t) + k_b y_b(t) + c_b\dot{y}_b(t) + F_c(t) = 0 \quad (2)$$

where $p(t) = y_a(t) - x(t)$ and the remaining variables are defined in Figure 1. The error signal measured at the error microphone is given by

$$e(t) = y_a(t - \delta_a) + y_b(t - \delta_b) \quad (3)$$

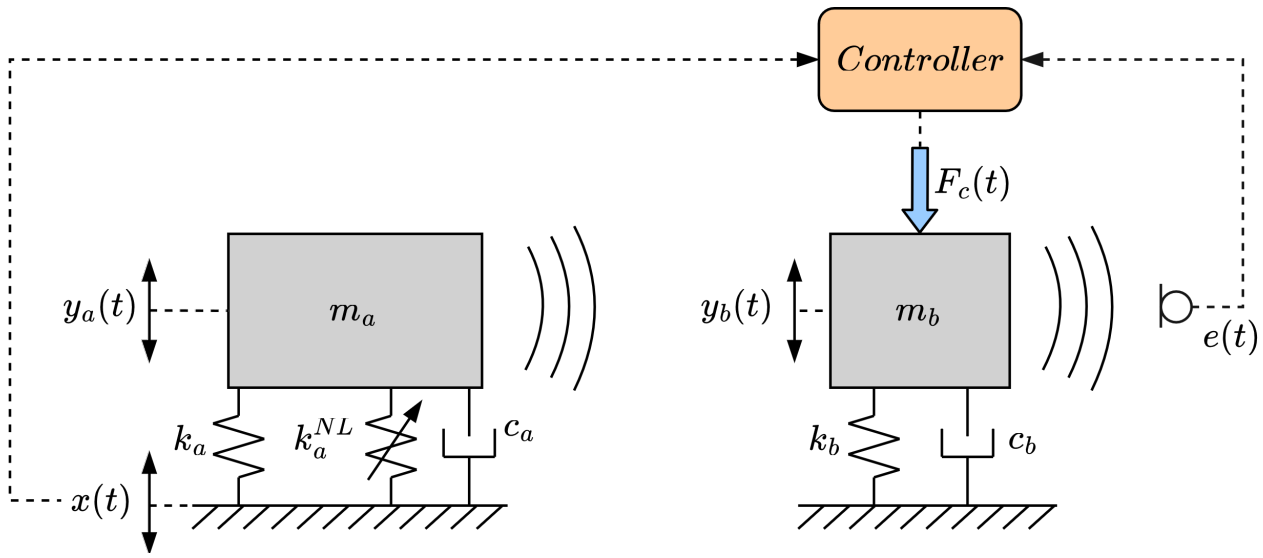

Figure 1. The simulated two-source system, where m_a is the mass of the primary oscillator, m_b is the mass of the secondary oscillator, $y_a(t)$ is the displacement of the primary oscillator, $y_b(t)$ is the displacement of the secondary oscillator, $x(t)$ is the displacement of the moving floor, k_a is the linear stiffness coefficient of the primary oscillator, k_a^{NL} is the nonlinear stiffness coefficient of the primary oscillator, k_b is the linear stiffness coefficient of the secondary oscillator, c_a is the damping coefficient of the primary oscillator, c_b is the damping coefficient of the secondary oscillator, and $F_c(t)$ is the control force applied to the secondary oscillator.

where δ_a and δ_b are the acoustic delays, in time, between the primary and secondary sources and the error microphone, respectively. In all cases, the signal $x[n]$ is white noise which has been passed through a low-pass filter with a cut-off frequency of 200 Hz.

2.2 Simulation method

The motion of the two masses are simulated in the time domain using the method of Antippa & Dubois [26]. This method compares to the Euler method, in which discrete systems may be simulated in the time domain by calculating accelerations in the current timestep from displacements and velocities in the previous timesteps, then updating the velocities in the current timestep from the calculated accelerations, and displacements from the calculated velocities. Unfortunately, this method is demonstrably unstable even if the system is unforced. Antippa & Dubois suggested that, by reordering the calculations of acceleration, velocity and displacement, simulation stability can be improved. For the simulated system here, the accelerations, velocities and displacements of the two masses are calculated iteratively according to equations (4-9).

$$\dot{y}_a[n] = \dot{y}_a[n-1] + \ddot{y}_a[n-1]\Delta t \tag{4}$$

$$y_a[n] = y_a[n-1] + \dot{y}_a[n]\Delta t \tag{5}$$

$$\dot{y}_b[n] = \dot{y}_b[n-1] + \ddot{y}_b[n-1]\Delta t \tag{6}$$

$$y_b[n] = y_b[n-1] + \dot{y}_b[n]\Delta t \tag{7}$$

$$\ddot{y}_a[n] = \frac{1}{m_a}(-k_a p[n] - k_a^{NL} p^3[n] - c_a \dot{p}[n]) \tag{8}$$

$$\ddot{y}_b[n] = \frac{1}{m_b}(-k_b y_b[n] - c_b \dot{y}_b[n] + F_c[n]) \tag{9}$$

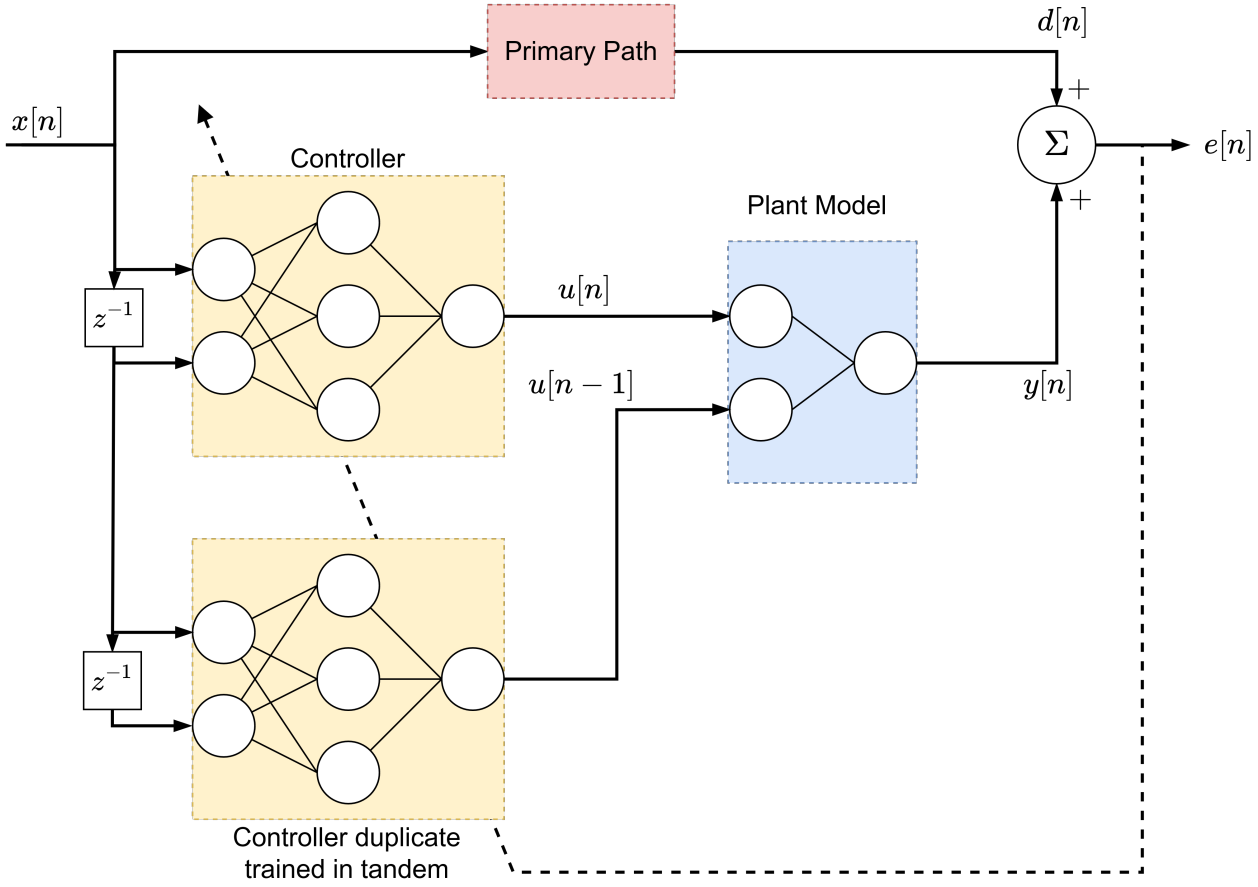

Figure 2. The dual-network architecture used to train the controller

where $p[n] = y_a[n] - x[n]$ and $\dot{p}[n] = \dot{y}_a[n] - \dot{x}[n]$.

3. CONTROLLER DESIGN AND TRAINING

3.1 Controller training architecture

A simplified diagram of the NN architecture used for the controller training is shown in Figure 2. Similarly to the case of a linear controller, the NN controller and plant model each take as input a tapped delay line of $x[n]$ or $u[n]$, respectively. The plant model output $y[n]$ is linearly summed with the disturbance signal $d[n]$ to generate the error signal $e[n]$ which is used to update the weights and biases of the controller NN via backpropagation through the full network. Given a tapped delay line of length L of the reference signal $x[n]$, given by

$$\mathbf{x}[n] = \begin{bmatrix} x[n] \\ x[n-1] \\ \vdots \\ x[n-L+1] \end{bmatrix} \tag{10}$$

the control signal $u[n]$ can be generated by inputting $\mathbf{x}[n]$ to the controller NN. If the NN architecture is that of a Multilayer Perceptron (MLP) with a single hidden layer, then the output of the NN is given by

$$u[n] = \sum_i w_i^{c,o} h_i^c + b^{c,o} \tag{11}$$

where $w_i^{c,o}$ are the output weights of the NN, $b^{c,o}$ is the NN output bias, and h_i^c are the NN hidden layer node values, given by

$$h_i^c = \sigma^c([\mathbf{W}\mathbf{x}[n]]_i + b_i^{c,h}) \tag{12}$$

where $\mathbf{W}$ is a matrix of weights between the input layer and hidden layer, $[\mathbf{W}\mathbf{x}[n]]_i$ is the i^{th} element of the vector $\mathbf{W}\mathbf{x}[n]$, $\sigma^c(\cdot)$ is the nonlinear activation function applied to the controller hidden layer, and $b_i^{c,h}$ is the bias of the i^{th} hidden layer node. In total,

$$u[n] = \sum_i w_i^{c,o} \sigma^c([\mathbf{W}\mathbf{x}[n]]_i + b_i^{c,h}) + b^{c,o} \tag{13}$$

However, a full tapped delay line $\mathbf{u}[n]$ is required to input to the plant model to generate the plant model output $\mathbf{y}[n]$ and therefore the error $\mathbf{e}[n]$. It is therefore necessary to have generate control signal values $u[n-1]$, $u[n-2]$, $\ldots$, $u[n-I+1]$ for a tapped delay line of length I. One solution to this problem is to store the values of the control signal in memory, and call upon them when evaluating the output of the plant model and updating the controller weights and biases. This approach has been used previously [27]. However, this approach means that the error signal $e[n]$ does not accurately reflect the control performance of the current iteration of the controller - it is calculated from the outputs of the current and previous $L-1$ iterations of the controller. This could plausibly lead to stability and performance issues in the training of the controller NN. As illustrated in Figure 2, an alternative approach is proposed here where the previous controller outputs $u[n-k]$ are generated as if the current iteration of the controller had always been in place. In general,

$$u[n-k] = \sum_i w_i^{c,o} \sigma^c([\mathbf{W}\mathbf{x}[n-k]]_i + b_i^{c,h}) + b^{c,o} \tag{14}$$

where all weights and biases in equation 14 are those of the current iteration of the controller during training. Irrespective of whether the values of $u[n-k]$ are called from memory or generated from the current iteration of the controller, standard backpropagation techniques can be used to update the weights and biases of the controller to minimise a given cost function of $e[n]$. This approach is clearly more computationally intensive than simply storing $u[n]$ in memory. However, computing $u[n]$ is only required during the training of the controller. The controller NN is assumed here to be fixed during operation and so, for the same sized NN controller, the computational cost to produce $u[n]$ from $x[n]$ in operation is independent of the training method.

3.2 Controller training details

The controller NN was trained to minimise the cost function defined as

$$J(e[n]) = e^2[n]. \tag{15}$$

The backpropagation method used was the ADAM method [28] with parameters $\alpha = 0.0001$, $\beta_1 = 0.9$, $\beta_2 = 0.99$, and $\epsilon = 10^{-7}$. All networks were trained for 2500 epochs with a batch size of 128 using 30 seconds of simulated data at a sample rate of 2 kHz. An L2 norm regularisation was applied to the controller hidden layer weights with a loss given by $l_2 \sum_i (h_i^c)^2$ with $l_2 = 0.0003$, which was found to give good generalisation without compromising performance greatly.

In all cases, the plant model used was an FIR filter (equivalent to an MLP with no hidden layer), which was capable of achieving high levels of modelling accuracy

due to the linear nature of the simulated secondary acoustic source.

4. RESULTS

4.1 Effect of training range

In a given application, the range of magnitudes of excitations of the nonlinear system which generates the disturbance will vary over a finite range. Over this range, the behaviour of the nonlinear system will vary in terms of the spectrum and statistics of its response. In this section, a NN controller with 30 hidden nodes is trained over a finite range of magnitudes of $x[n]$. The controller weights and biases are then fixed, and its performance is tested, in simulation, at a set of excitation magnitudes. The testing magnitudes are defined over a range of $[0, 10^{-5}]$ m.

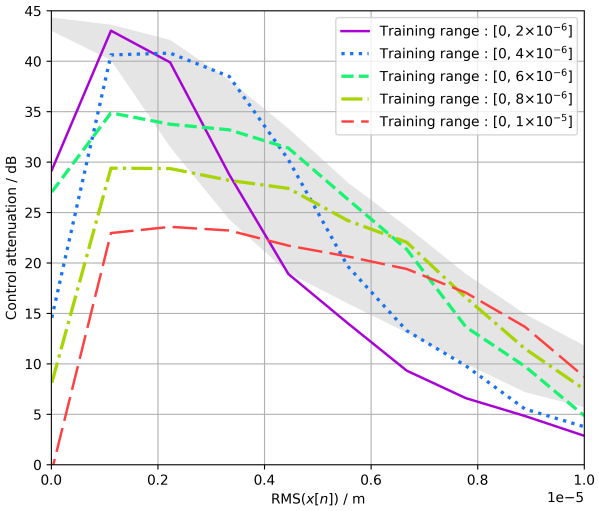

Figure 3. Performance of the NN controller as training width varies. The lower boundary of the grey region represents the maximum simulated control performance achieved using the FxLMS algorithm. The upper boundary represents the maximum control performance achieved by a NN with 30 hidden nodes trained and tested at a single magnitude of $x[n]$.

Figure 3 shows the control performance of 5 NN controllers trained over different excitation ranges. As a result of their finite size, the networks have limited modelling capacity. Therefore, none of the individual networks is capable of achieving optimal performance over a wide range of excitation magnitudes. Compared to the performance of the FxLMS algorithm, the networks generally achieve the highest level of performance just below the upper limit of their training range, with the relative performance dropping off as the magnitude of $x[n]$ is increased or decreased. This may be partly explained by the cost function used in training, given by Equation (15). This cost function is not normalised with respect to the training inputs to the NN controller. Therefore, the training algorithm will prioritise controlling the largest disturbances within its training set, which correspond to disturbances with magnitudes at, or just below, the upper bound of the training range. It is also notable that the networks tend to perform poorly at very low magnitudes of input to the nonlinear system. This may similarly be a result of the chosen cost function. However, it should be noted that small inputs to a nonlinear system will generally correspond to small outputs. As such, control of the most linear behaviour of the system is unlikely to be important, as the disturbances caused by low magnitude inputs will themselves be small.

4.2 Frequency and time domain performance

Figure 4 presents an example of the control performance of an NN controller compared to the FxLMS algorithm in the frequency domain. The equivalent time domain results are presented in Figure 5. The NN controller has 30 hidden layer nodes and was trained over the range $[0, 6 \times 10^{-6}]$ m. The controllers were tested at an input magnitude of 5×10^{-6} m. The linear FxLMS controller achieves a total attenuation of 16.4 dB, wheras the NN controller achieves a total attenuation of 28.0 dB. Both controllers achieve broadband control up to 200 Hz with similar shaped error spectra, with the NN controller achieving 10-15 dB greater performance across this entire range.

4.3 Effect of network size

It is well known that increasing the number of hidden nodes in a NN can improve its capacity for generalisation in a given task, up to the point where the network begins to overfit. In this section, the effect of changing the number of hidden nodes in the controller NN on the control performance is investigated when the training range is fixed.

Figure 6 shows the control performance of 6 networks, with a range of number of hidden layer nodes, trained over the same range of excitation magnitudes. The

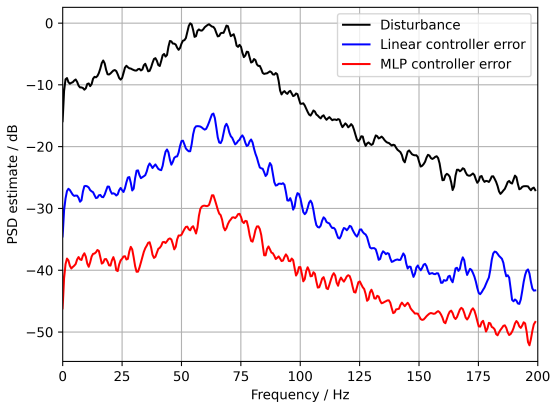

Figure 4. Performance of an NN controller and FxLMS linear controller in the frequency domain

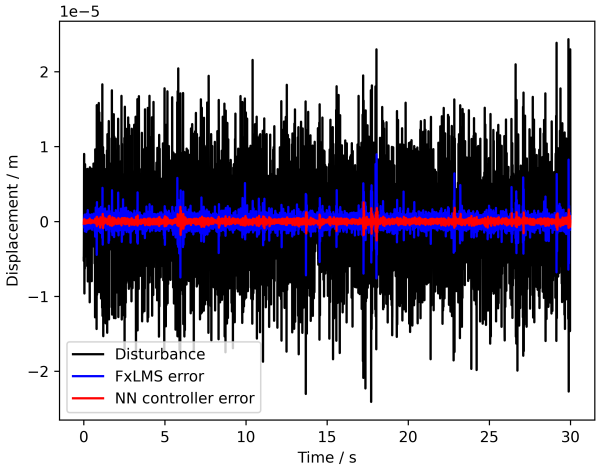

Figure 5. Performance of an NN controller and FxLMS linear controller in the time domain

region over which the networks were trained is shown by the red region, and the grey region represents the same performance envelope as described in Figure 3. As discussed in the previous section, all networks achieve the highest level of performance compared to FxLMS just below the upper bound of the training range. Within the training range, with the exception of very low input magnitudes, increasing the number of hidden nodes within the controller NN increases control performance. Above the training range, the networks still tend to generalise quite well, with networks with greater than or equal to 5 hidden nodes still consistently performing at least as well as FxLMS. However, the results also show that increasing the number of hidden nodes in the NN does not necessarily improve generalised performance in this region.

4.4 Performance under changes input magnitude

In the previous sections, the performance of the trained controllers is tested at fixed magnitudes of $x[n]$. However, in a real application, the magnitude of the excitation may vary over time. Therefore, in this section, the performance of a given trained NN controller is tested as the excitation magnitude is varied over time within a range where the controller is known to have acceptable performance.

Figure 7 shows the performance, in the time domain, of a NN controller with 30 hidden nodes which has been trained in the range $[0, 6 \times 10^{-6}]$ m tested in a simulation where the magnitude of the input to the nonlinear system switches instantaneously between a level of 3×10^{-6} m and 5×10^{-6} m. In each simulation, the switching occurs at $t = 5$ s. The residual error from the NN controller is not dramatically affected by sudden changes in the magnitude of the input compared to control under a constant magnitude input.

5. CONCLUSIONS

In this paper, the performance of neural network-based feedforward active acoustic control has been compared to the FxLMS algorithm in a set of time-discrete simulations. The generalised performance of the controllers across a range of system excitation magnitudes has been investigated. It has been shown that the fixed NN controllers have the ability to generalise well across a range of magnitudes of excitations of the simulated nonlinear system. Relative to the performance of the FxLMS algorithm, the control performance of the NN controllers is generally highest close to the upper limit of the range of magnitudes of excitation over which the controllers are trained. It is speculated that this is a result of the chosen cost function during the training of the NNs. Increasing the training range generally improves performance at higher excitation magnitudes, but compromises performance at lower magnitudes. Increasing the number of hidden nodes in the NN controllers improves performance within the trained range, with some evidence that generalised performance is improved above the trained range. The residual error of a NN controller under sudden changes to the system

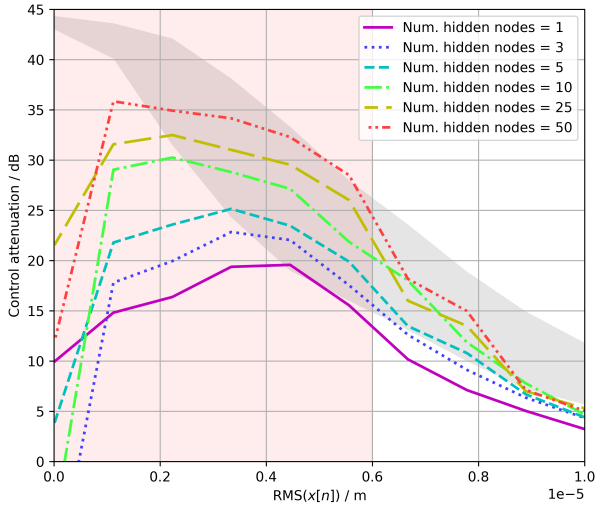

Figure 6. Performance of the NN controller as the number of hidden layer nodes is varied

excitation level has been investigated, with the controller showing robust behaviour over this transition.

6. ACKNOWLEDGMENTS

This work was supported by an EPSRC Prosperity Partnership (No. EP/S03661X/1). The authors acknowledge the use of the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton, in the completion of this work.

7. REFERENCES

[1] X. Pike and J. Cheer, "Limitations of FxLMS in feedforward active vibration control of a nonlinear two-degree-of-freedom system," *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, vol. 265, pp. 5219–5229, Feb. 2023.

[2] M. Costa, J. Bermudez, and N. Bershad, "Stochastic analysis of the filtered-x LMS algorithm in systems with nonlinear secondary paths," *IEEE Transactions on Signal Processing*, vol. 50, pp. 1327–1342, June 2002.

[3] O. Tobias and R. Seara, "On the LMS algorithm with constant and variable leakage factor in a nonlinear environment," *IEEE Transactions on Signal Processing*, vol. 54, pp. 3448–3458, Sept. 2006.

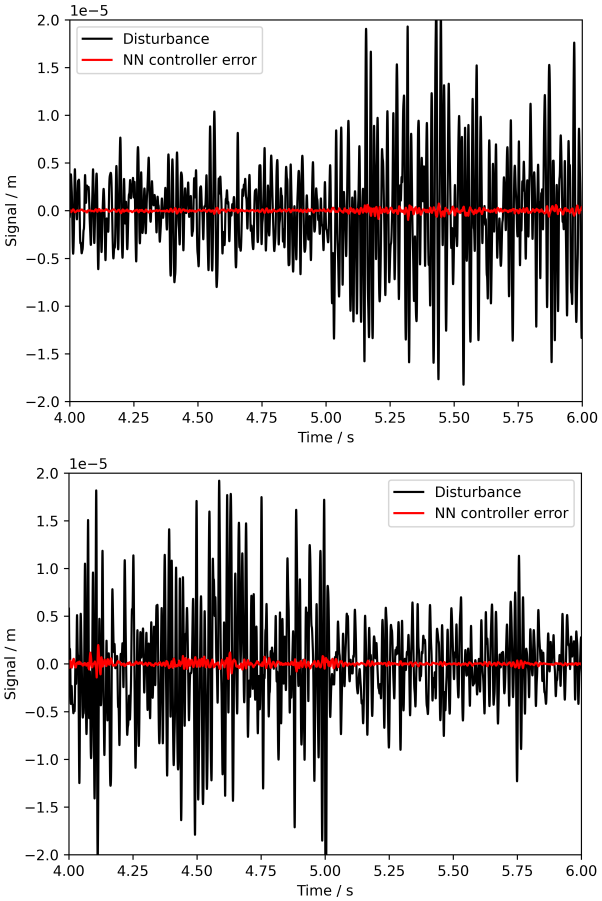

Figure 7. Performance of the NN controller when the magnitude of the system input changes suddenly

[4] S. Snyder and N. Tanaka, "Active control of vibration using a neural network," *IEEE Transactions on Neural Networks*, vol. 6, pp. 819–828, July 1995.

[5] L. Tan and J. Jiang, "Adaptive volterra filters for active control of nonlinear noise processes," *IEEE Transactions on Signal Processing*, vol. 49, no. 8, pp. 1667–1676, 2001.

[6] E. P. Reddy, D. P. Das, and K. Prabhu, "Fast exact multichannel FSLMS algorithm for active noise control," *Signal Processing*, vol. 89, pp. 952–956, May 2009.

[7] C. Wangler and C. Hansen, "Genetic algorithm adaptation of non-linear filter structures for active sound and vibration control," in *Proceedings of ICASSP '94.*

IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE.

[8] Q.-Z. Zhang and W.-S. Gan, "Active noise control using a simplified fuzzy neural network," *Journal of Sound and Vibration*, vol. 272, pp. 437–449, Apr. 2004.

[9] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, Jan. 1989.

[10] M. Bouchard, B. Paillard, and Chon Tan Le Dinh, "Improved training of neural networks for the nonlinear active control of sound and vibration," vol. 10, no. 2, pp. 391–401.

[11] K. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," vol. 1, no. 1, pp. 4–27.

[12] D. Hong, H. Lee, Y. Han, and B. Kim, "Numerical feasibility study for transverse vibration control of rotating shaft with a neural network-based tracking algorithm," *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, vol. 263, pp. 1293–1298, Aug. 2021.

[13] H.-S. Kim, "Development of seismic response simulation model for building structures with semi-active control devices using recurrent neural network," vol. 10, no. 11, p. 3915.

[14] C.-Y. Chang and F.-B. Luoh, "Enhancement of active noise control using neural-based filtered-x algorithm," vol. 305, no. 1, pp. 348–356.

[15] Y. Yan, L. Dong, Y. Han, and W. Li, "A general inverse control model of a magneto-rheological damper based on neural network," vol. 28, no. 7, pp. 952–963.

[16] Kyungmin Na and Soo-Ik Chae, "Single-sensor active noise cancellation using recurrent neural network predictors," in *Proceedings of International Conference on Neural Networks (ICNN'97)*, vol. 4, pp. 2153–2156, IEEE.

[17] C. Chen and Tzi-Dar Chiueh, "Multilayer perceptron neural networks for active noise cancellation," in *1996 IEEE International Symposium on Circuits and Systems. Circuits and Systems Connecting the World. ISCAS 96*, vol. 3, pp. 523–526, IEEE.

[18] M. Salmasi, H. Mahdavi-Nasab, and H. Pourghassem, "Comparison of feed-forward and recurrent neural networks in active cancellation of sound noise," in *2011 International Conference on Multimedia and Signal Processing*, pp. 25–29, IEEE.

[19] T. Matsuura, T. Hiei, H. Itoh, and K. Torikoshi, "Active noise control by using prediction of time series data with a neural network," in *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, vol. 3, pp. 2070–2075, IEEE.

[20] K. Hiramoto and T. Matsuoka, "Active vibration control of structural systems with a preview of a future seismic waveform generated by remote waveform observation data and an artificial intelligence–based waveform estimation system," vol. 26, no. 17, pp. 1602–1613.

[21] Q. Wang, J. Wang, X. Huang, and L. Zhang, "Semi-active nonsmooth control for building structure with deep learning," vol. 2017, pp. 1–8.

[22] J. Liu, X. Li, X. Zhang, and X. Chen, "Modeling and simulation of energy-regenerative active suspension based on BP neural network PID control," vol. 2019, pp. 1–8.

[23] H. Zhang and D. Wang, "Deep ANC: A deep learning approach to active noise control," vol. 141, pp. 1–10.

[24] H. Zhang and D. Wang, "Deep MCANC: A deep learning approach to multi-channel active noise control," vol. 158, pp. 318–327.

[25] R. Ranjan, J. He, T. Murao, L. Bhan, and W. S. Gan, "Selective active noise control system for open windows using sound classification,"

[26] A. F. Antippa, D. M. Dubois, and D. M. Dubois, "Time-symmetric discretization of the harmonic oscillator," in *AIP Conference Proceedings*, AIP, 2010.

[27] Z.-c. Qiu and W.-z. Zhang, "Trajectory planning and diagonal recurrent neural network vibration control of a flexible manipulator using structural light sensor," vol. 132, pp. 563–594.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.