



**HAL**  
open science

# Deep reinforcement learning for tuning active vibration control on a smart piezoelectric beam

Maryne Febvre, Jonathan Rodriguez, Simon Chesne, Manuel Collet

► **To cite this version:**

Maryne Febvre, Jonathan Rodriguez, Simon Chesne, Manuel Collet. Deep reinforcement learning for tuning active vibration control on a smart piezoelectric beam. *Journal of Intelligent Material Systems and Structures*, 2024, 35 (14), pp.1149-1165. 10.1177/1045389X241260976 . hal-04770230

**HAL Id: hal-04770230**

**<https://hal.science/hal-04770230v1>**

Submitted on 12 Nov 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.


L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

---

DOI: <https://doi.org/10.1177/1045389X241260976>


## DEEP REINFORCEMENT LEARNING FOR TUNING ACTIVE VIBRATION CONTROL ON A SMART PIEZOELECTRIC BEAM.

MARYNE FEBVRE 

INSA Lyon, CNRS, LaMCoS, UMR5259, 69621 Villeurbanne, France and  
CNRS, Ecole Centrale de Lyon, ENTPE, LTDS, UMR5513, 69134 Ecully, France

JONATHAN RODRIGUEZ 

INSA Lyon, CNRS, LaMCoS, UMR5259, 69621 Villeurbanne, France

SIMON CHESNE 

INSA Lyon, CNRS, LaMCoS, UMR5259, 69621 Villeurbanne, France

MANUEL COLLET 

CNRS, Ecole Centrale de Lyon, ENTPE, LTDS, UMR5513, 69134 Ecully, France

*Version July 2024*

### Abstract

Piezoelectric transducers are used within smart structures to create functions such as energy harvesting, wave propagation or vibration control to prevent human discomfort, material fatigue, and instability. The design of the structure becomes more complex with shape optimization and the integration of multiple transducers. Most active vibration control strategies require the tuning of multiple parameters. In addition, the optimization of control methods has to consider experimental uncertainties and the global effect of local actuation. This paper presents the use of a Deep Reinforcement Learning (DRL) algorithm to tune a pseudo lead-lag controller on an experimental smart cantilever beam. The algorithm is trained to maximize a reward function that represents the objective of vibration mitigation. An experimental model is estimated from measurements to accelerate the DRL's interaction with the environment. The paper compares DRL tuning strategies with  $H_2$  and  $H_\infty$  norm minimization approaches. It demonstrates the efficiency of DRL tuning by comparing the control performance of the different tuning methods on the model and experimental setup.

*Keywords:* Active Control, Vibration Control, Feedback Control, Machine Learning, Neural Network, Cantilever Beam, Metamaterial, Parameter Estimation, Smart Structures, Piezoelectric Transducer

### 1. INTRODUCTION

Since the discovery of the piezoelectric effect by the Curie brothers in the 19th century (Curie (1984)), piezoelectric materials have found diverse applications. They are employed in various roles, from sensors for monitoring to actuators for controlling structural vibrations. The control of vibrations is paramount for ensuring the safety, functionality, and longevity of structures from bridges to space engines, as well as for preserving the comfort and well-being of individuals within and around these structures. In contemporary applications, sensors and actuators can be integrated directly into structures using multi-physical materials often referred to as "smart" materials, thereby imbuing structures with additional functionalities. As part of smart materials, piezoelectric transducers transfer mechanical energy into electrical energy with a reciprocity effect (M.G.Lippmann (1881)) capable of acting as actuators or sensors and contributing to the creation of a smart structures capable of autonomous control.

In the context of vibration control applications, various active methods have been developed that involve integrating smart materials directly into structures: PID control (Ziegler and Nichols (2022), Khot *et al.* (2011), Jovanović *et al.* (2013)), optimal control (Foutsitzi *et al.* (2003), Stavroulakis *et al.* (2005)) and robust control (Tani *et al.* (1995), Doyle *et al.* (1989), Rodriguez *et al.* (2022)). In general, these applications consider a limited amount of transducers. However, optimizing all control parameters becomes increasingly challenging as the system's dimensions and complexity grow, particularly when accounting for physical uncertainties.

To address this issue, researchers have recently explored intelligent control (Li *et al.* (2009)) in light of advancements in computational hardware. Intelligent control defines both fuzzy logic and artificial intelligence control. Fuzzy logic is a control strategy based on heuristic decision rules with "if then" structure, which is a generalization of boolean logic operators. Fuzzy rules are processed in parallel and in real-time and the operation should be fast enough to compute the output in one sampling period. In order to maintain the real time efficiency of the method in experimental

conditions, the number of fuzzy rules must be limited (Kwak and Sciulli (1996)) since the algorithm has to choose at each time step the suitable controller.

Artificial intelligence-based control aims to replicate the human brain by utilizing a Neural Network (NN) representation (Rosenblatt (1958)). Lee.G (Lee (1996)) introduced the use of NNs on a smart cantilever beam experiment for system identification, system state estimation or vibration control with the so-called Artificial Neural Network (ANN) controller. Jha and Rower (Jha and Rower (2002)) conducted an experiment on a smart cantilever beam to test the usage of an ANN controller and its robustness against parametric uncertainties with multiple disturbance signals. ANNs are also utilized as predictive models for feed-forward control to compensate for the hysteresis behavior of piezoelectric elements (Liang *et al.* (2019)) or to attenuate disturbances on a cantilever beam (D.Snyder and Tanaka (1995)) and other smart composite structures (Smyser and Chandrashekhara (1997), Mohit *et al.* (2015)). The works highlight a lack of consistency and predictability in performance due to the use of neural network controllers that are trained offline using Supervised Learning with a restrictive database, simulated by a model or measured on experimental setups.

According to recent advancements in Artificial Intelligence algorithms (Cardon *et al.* (2018)) and hardware technology (Lee (2021)), new Machine Learning methodologies using NNs can be used to control systems. Considering Multiple Input Multiple Output (MIMO) systems, Unmanned Aerial Vehicles (UAV) have been recently controlled using NNs trained with Reinforcement Learning (RL) algorithm without the need for a database. RL controllers relevance compared to PID controller has been proven experimentally on trajectory tracking experiment on UAV (Koch *et al.* (2019)). The performance of RL controllers is dependent on the definition of the observation space, as demonstrated in the lift control experiment of a fixed-wing vehicle by (Guerra-Langan *et al.* (2022)). In this experiment, various RL controllers are compared to a manually tuned PID controller. The RL algorithms are trained on a model which does not take into account noise and experimental uncertainties. Alternatively, training the RL on an experimental setup can be time-consuming depending on the complexity of the system, as shown in (Haughn *et al.* (2024)).

Qiu *et al.* (2021) use a Reinforcement Learning (RL) controller to control a plate equipped with 5 piezoelectric transducers (4 for actuation and 1 for measurement) and minimize its response to sinusoidal excitation at resonant frequencies. Simulation results show that the RL controller is faster than a classical PD controller in damping the modal response. However, with the experimental setup, the RL controller has slower performance for transient vibrations but achieves similar performance after a few periods, depending on the frequency. All of these studies use RL to provide the time-domain control signal.

More recent works have also used RL to set controller parameters. For example, Khalatbarisoltani (2019) works with an active mass drive (AMD) system on structures stressed with seismic events. Reinforcement learning is used with online tuning to determine the gains of a fuzzy PD controller. Experimental tests allow to prove successful amplitude attenuation of the structural response amplitude in response to seismic perturbations. Pisarski and Jankowski (2022) develop a switching control policy on a numerical Euler Bernoulli model of a multi-sensor beam monitor with semi-active control. A reinforcement learning method based on structure cost function minimization is implemented to adjust the controller parameters. By comparing with other methods, the results show the validation of the RL-based semi-active control method for transient vibration mitigation on a numerical model. Panda *et al.* (2024) propose to use Reinforcement Learning algorithm with policy gradient based method to tune a P and a PI controller. The method's efficiency is illustrated on two numerical cases considering harmonic excitations: a quarter car model with active suspension system and an 8-story benchmark building. According to the research cited above, Reinforcement Learning strategies have two main uses: as a controller that generates the control signal directly in the time domain, or to adjust the parameters of an existing controller.

In line with the recent literature, this article explores the use of RL algorithms to adjust controller parameters and mitigate different perturbation signals on an experimental smart cantilever. The purpose is to introduce the Deep Reinforcement Learning (DRL) tool to an active vibration control problem, starting with a structure with two piezoelectric transducers.

The DRL optimization will be linked to the cantilever beam by defining its Environment, State, Action and Reward functions based on vibration domain metrics. Here, the Neural Network (NN) is used to tune the three parameters of a pseudo lead lag controller. Hence, the computation of the NN output is only necessary during the training. Deep Neural Networks (DNN) are able to relate efficiently inputs and outputs considering high dimension MIMO systems (Nguyen *et al.* (2023)). Using this method in a vibration control problem can be useful to design distributive control considering piezoelectric transducers within a network. The Trust Region Policy Optimization (TRPO) for NN guarantees monotonous improvements during training (Schulman *et al.* (2015)), thus staying close to the stability domain of the controlled structure. A pseudo lead lag control architecture is justified since spillover instabilities can occur due to residual modes (Preumont (2018)) close to the controller target bandwidth. It can be solved by using optimal sensor placement (Khushnood *et al.* (2016)), and increasing the order of the controller (Jovanović *et al.* (2014)). Compared to simple derivative feedback (Febvre *et al.* (2023)), the use of a pseudo lead-lag controller allows to reduce the spillover effect in the tuning process. To decrease NN training time, a model of the experimental setup is built from measurements with poles and zeros estimation. Using a model during the training process avoids waiting for the acquisition time on the real structure.

Controllers are also tuned using  $H_2$  and  $H_\infty$  norm minimization, which serves as the reference controller. The performance of the controllers is tested on the model and measured in the experimental setup. Comparison of DRL with norm minimization demonstrates the efficiency and reliability of the DRL method in tuning a stable lead-lag

controller for a smart cantilever beam.

## 2. METHODOLOGIES

In the following section, a control strategy is implemented on a smart structure with one piezoelectric actuator and two sensors. This section introduces tuning concepts based on norm minimization and Deep Reinforcement Learning.

### 2.1. Control Strategy

A feedback control law  $C(s)$  is implemented on the system  $G(s)$  which is the smart structure to be controlled, as defined in the block diagram Figure 1. In this architecture,  $r = 0$  is the target signal,  $e$  the error,  $v_d$  the disturbance,  $v_c$  the command,  $v$  the input actuation signal of the system  $G(s) = [G_1(s); G_2(s)]$  and  $[x_1; x_2]$  the respective system outputs. Both the input disturbance and the control signals are applied to the system through one channel  $v$ .

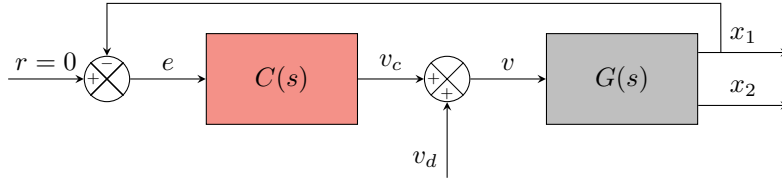


FIG. 1.—: Control loop block diagram

A parametric approach is performed to tune the control law parameters with criterion minimization based on the closed-loop transfer functions  $G_{c,1}(s)$  and  $G_{c,2}(s)$ , such as:

$$G_{c,1}(s) = \frac{G_1(s)}{1 + G_1(s)C(s)} \quad G_{c,2}(s) = \frac{G_2(s)}{1 + G_1(s)C(s)} \quad (1)$$

The first criterion chosen for the minimization is based on the  $H_2$  norm of the closed-loop transfer function restricted to a finite frequency band, such as:

$$\|G_{c,q}(j\omega)\|_2 = \sqrt{\frac{1}{2\pi} \int_{\omega_1}^{\omega_2} G_{c,q}(j\omega)^2 d\omega} \quad (2)$$

with  $j \in \mathbb{C}$  the imaginary variable and  $\omega$  the frequency in  $[rad/sec]$  and  $q \in [1; 2]$ .

The second criterion is defined with the  $H_\infty$  norm in the case of linear systems within the same frequency band, such as:

$$\|G_{c,q}(j\omega)\|_\infty = \max(G_{c,q}(j\omega)) \quad (3)$$

Different disturbance signals have to be considered to obtain equivalent metrics in the time domain. The first disturbance signal  $v_{d1}(t)$  is a random noise  $Rnd(t)$  with a sample frequency  $F_s$  in Hz and a second-order filter with a cutoff frequency  $F_c$  in Hz such as:

$$v_{d1}(t) = Rnd(t) * L^{-1} \left\{ \left( \frac{2\pi F_c}{s + 2\pi F_c} \right)^2 \right\} \quad (4)$$

with  $L^{-1}$  the inverse Laplace transform and  $*$  being the temporal convolution. The root mean square of the system outputs, disturbed by this band-limited noise signal, represents the  $H_2$  norm in the time domain.

The second signal  $v_{d2}(t)$  is a chirp with a sample frequency  $F_s$  starting from  $f_0$  to  $f_1$  in Hz at time  $t_f$  in seconds, such as:

$$v_{d2}(t) = V_{d2} \sin(\phi(t)) \quad (5)$$

$$f(t) = \frac{1}{2\pi} \frac{d\phi(t)}{dt} = f_o + \frac{(f_1 - f_0)}{t_f} t \quad (6)$$

with  $\phi(t)$  the phase considering  $\phi(0) = 0$  and  $V_{d2} \in \mathbb{R}$  the constant amplitude. The maximum absolute value of the system outputs, disturbed by this chirp signal, is representative of  $H_\infty$  norm in the time domain. In the result section, the metrics are used to compare controlled results.

### 2.2. Deep Reinforcement Learning Algorithm

Reinforcement Learning (RL) methods (Sutton and Barto (1992)) are Machine Learning approaches without any database requirement. During the training, the algorithm learns on a defined environment with a trial-error process by trying to improve a defined Reward  $R$  as displayed in Figure 2. The training is managed by an Agent who takes as input, the States  $S$  from the environment and chooses the Actions  $A_c$  to run accordingly. As part of RL, Markov

Decision process with policy based method defines the Agent with two NN, an Actor with a conditional function  $\pi$  and its parameters  $A_c$ ,  $S$  and  $\theta$  and a Critic function  $V$  and its parameters  $S$  and  $\psi$ . The  $\theta$  and  $\psi$  variables define NN parameters.

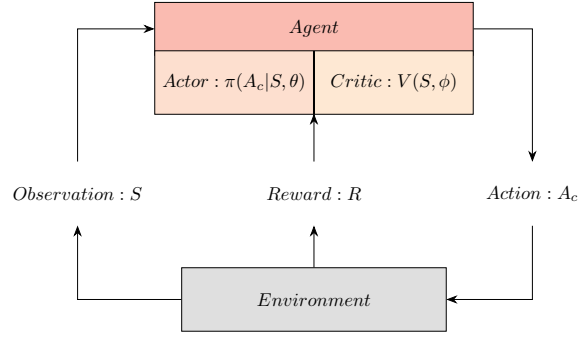


FIG. 2.—: Reinforcement Learning algorithm: scheme

Deep Reinforcement Learning strategy uses Neural Networks (NNs) in the Agent as a non-linear function to optimize according to the reward definition. Made by assembling perceptrons (Rosenblatt (1958)), NNs are defined with layers differentiated into three parts: the input layer with  $N_i \in \mathbb{N}$  input neurons, the hidden layers with  $N_{hu} \in \mathbb{N}$  hidden neurons per layer and  $N_l \in \mathbb{N}$  layers and the output layer with  $N_o \in \mathbb{N}$  number of neurons. Figure 3 is a schematic representation of a NN.

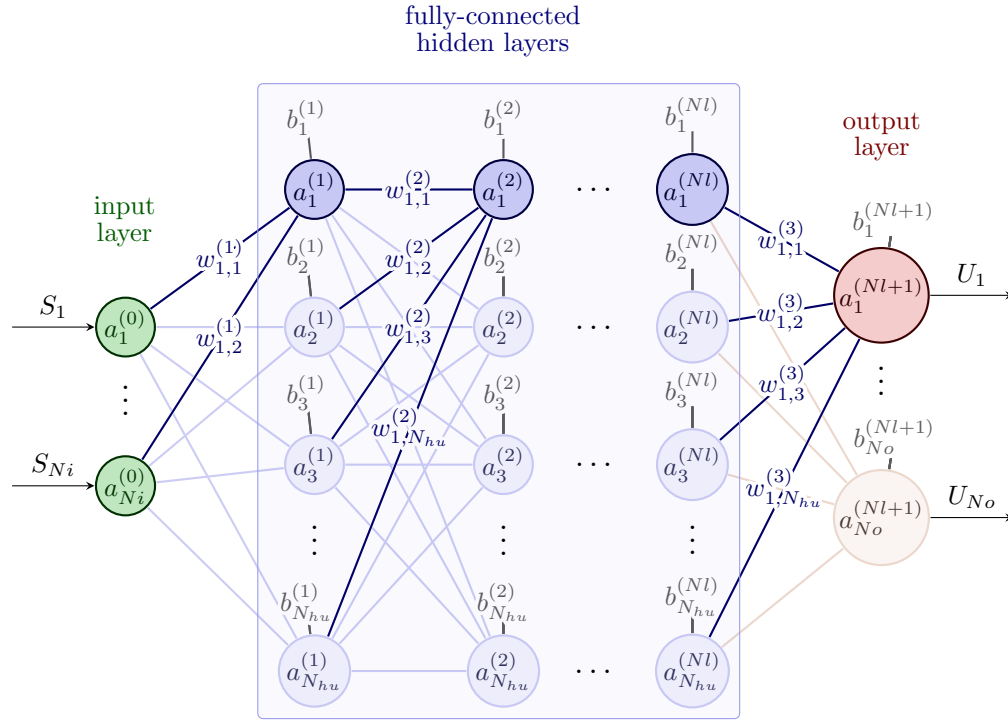


FIG. 3.—: Neural Network schematic representation

Each layer  $l \in \mathbb{N}$  is composed of independent neurons defined with a bias  $b^{(l)} \in \mathbb{R}$  and an activation function  $f_a^{(l)} \in \mathbb{R}^{n_{l-1}} \rightarrow \mathbb{R}^{n_l}$ . The connection between neurons in one layer and another is determined by a weighting parameter denoted as  $w^{(l)} \in \mathbb{R}$ . The output value  $a_{n_l}^{(l)} \in \mathbb{R}$  of the neuron  $n_l \in \mathbb{N}$  in the layer  $l$  is computed such as:

$$\begin{bmatrix} a_1^{(l)} \\ a_2^{(l)} \\ \vdots \\ a_{n_l}^{(l)} \end{bmatrix} = f_a^{(l)} \left( \begin{bmatrix} w_{1,1}^{(l)} & w_{1,2}^{(l)} & \cdots & w_{1,n_{l-1}}^{(l)} \\ w_{2,1}^{(l)} & w_{2,2}^{(l)} & \cdots & w_{2,n_{l-1}}^{(l)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n_l,1}^{(l)} & w_{n_l,2}^{(l)} & \cdots & w_{n_l,n_{l-1}}^{(l)} \end{bmatrix} \begin{bmatrix} a_1^{(l-1)} \\ a_2^{(l-1)} \\ \vdots \\ a_{n_{l-1}}^{(l-1)} \end{bmatrix} + \begin{bmatrix} b_1^{(l)} \\ b_2^{(l)} \\ \vdots \\ b_{n_l}^{(l)} \end{bmatrix} \right) \quad (7)$$

$$\mathbf{A}^{(l)} = f_a^{(l)}(\mathbf{W}^{(l)}\mathbf{A}^{(l-1)} + \mathbf{B}^{(l)}) \quad (8)$$

with  $\mathbf{A}^{(l)} \in \mathbb{R}^{n_l}$  the output vector of the layer  $l$ ,  $n_l \in \mathbb{N}$  the number of neurons in the layer  $l$ ,  $\mathbf{W}^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$  the weight matrix between the layers  $l-1$  and  $l$ ,  $\mathbf{B}^{(l)} \in \mathbb{R}^{n_l}$  the bias vector of the layer  $l$ . The NN parameters to adjust in the training process are the bias and the weights of each layer. The Trust Region Policy Optimization (TRPO) (Schulman *et al.* (2015)) is used to train these Neural Networks. The agent defines the action to run on the environment according to a policy and a model based on the environment. The policy is stored in a NN named the Actor  $\pi$ . It indicates which actions  $A$  to run on the environment according to the observations  $S$ , such as:

$$\pi(A|S, \theta) = \begin{cases} U_{i=\{1:\frac{N_o}{2}\}} = \mu = \tanh(a_i^{(N_{l+1})}) \\ U_{i=\{\frac{N_o}{2}+1:N_o\}} = \sigma = \tanh(\ln(1 + \exp(a_i^{(N_{l+1})}))) \end{cases} \quad (9)$$

With  $\theta$  the Actor's NN weights and bias,  $\mu$  the mean and  $\sigma$  the standard deviation of the Gaussian probability distribution for each continuous action.

This model-based method builds a representation of the environment with a NN named the Critic  $V$ . It gives an estimation of an expected reward  $R_V \in \mathbb{R}$  according to the observations  $S$  from the environment, such as:

$$V(S, \psi) = R_V \quad (10)$$

with  $\psi$  the critic's NN weights and bias. For both Critic and Actor NNs, a ReLU (Fukushima (1969)) function  $f_a$  is chosen for neurons activation function, such as:

$$f_a(y) = \begin{cases} y & y \geq 0 \\ 0 & y < 0 \end{cases}, \quad y \in \mathbb{R} \quad (11)$$

Main outlines of the TRPO algorithm updating process of  $\psi$  and  $\theta$  are described in Algorithm 1:

---

#### Algorithm 1 TRPO Algorithm

---

<b>Require:</b> $N \geq 0; R_{max}$	▷ End training conditions
<b>while</b> $t \leq \bar{N}$ <b>do</b> or $R_t \leq R_{max}$	
$A_c \leftarrow 0$	▷ Initial Action
$S_0 \leftarrow Env(A_c)$	▷ Get Environment State
$R_V \leftarrow V(S_0, \psi)$	▷ Compute Critic output
$A_c \leftarrow \pi(A_c S_0, \theta)$	▷ Compute Actor output
$R \leftarrow Env(A_c)$	▷ Get Environment Reward
$\theta_{new} \leftarrow \min(Loss_\pi(\theta))$	▷ Minimize Critic Loss
$\psi_{new} \leftarrow \min(Loss_V(\psi))$	▷ Minimize Actor Loss
$\theta \leftarrow \theta_{new}$	▷ Update Critic parameters
$\psi \leftarrow \psi_{new}$	▷ Update Actor parameters
$t \leftarrow t + 1$	
<b>end while</b>	
$A_c \leftarrow Actor(S_0)$	▷ Get end training Action

---

The agent's training is defined by two constraints:  $N \in \mathbb{N}$ , specifying the maximum number of episodes, and  $R_{max} \in \mathbb{R}$  indicating the maximum reward target to be achieved. An Episode starts with the environment's initial state  $S_0 = Env(A_c)$  defined here such as  $A_c = 0$ . The Critic estimates the expected reward  $R_V$  based on this initial state and the Actor chooses an Action  $A_c$ . The action is applied to the environment and Reward  $R$  can be computed to update  $\theta$  and  $\psi$  parameters with Critic and Actor loss functions minimization using a line search algorithm (Nocedal and Wright (1999)) such as:

$$Loss_V(\psi) = (R + \gamma V(S, \psi) - V(S, \psi))^2 \quad (12)$$

$$Loss_\pi(\theta) = -\frac{\pi(A|S, \theta_{new})}{\pi(A|S, \theta)} (R + \gamma V(S, \psi)) + w \frac{1}{2} \sum_{k=1}^P \ln(2\pi \cdot \exp(1) \cdot \sigma_k^2) \quad (13)$$

with  $P = N_o/2 \in \mathbb{N}$  the number of output actions,  $w \in \mathbb{R}$  the Entropy Loss Weigth,  $\gamma \in \mathbb{R}$  the Discount Factor,  $\sigma_k \in \mathbb{R}$  the standard deviation of the Gaussian probability distribution for each continuous action  $k \in \mathbb{N}$  and  $\exp()$  the exponential function.

This method adjusts NNs according to the case of vibration mitigation considering continuous actions cases. It is not necessary to define all possible actions, only the boundaries between the maximum and minimum values, which allows for a large number of output actions without enumerating all the combinations. This results in smaller NNs, reducing the necessary training time. Other policy gradient methods could also be used in this application case but TRPO is chosen for its robustness. Although it requires more computation than its simplified version Proximal Policy Optimization (PPO) (Schulman *et al.* (2017)), TRPO updates its policy within a trust region close to the current, thus avoiding drops in performance in most cases. DRL Observation, Action and Reward needs to be defined according to the vibration mitigation objective on the smart structure. This paper explores into the design of DRL components according to this application.

### 3. EXPERIMENTAL SETUP

This section first presents the setup used to perform the experimental implementation of the previously described methodology. Since the DRL training process needs multiple interactions with the environment, a model of the structure response is created to reduce the training time for the control optimization. Then, DRL elements are defined in order to tune a pseudo lead-lag controller that minimizes the vibrations of the selected smart cantilever beam.

#### 3.1. Setup Definition

The experimental setup is a cantilever beam with two collocated piezoelectric patches close to the clamped end as displayed in Figure 4. The beam has length  $L_b$ , width  $W_b$ , and thickness  $T_b$ . The two collocated piezoelectric elements are attached to the beam at a distance  $D_p$  from the supported end. The piezoelectric transducers dimensions are length  $L_p$ , width  $W_p$ , and thickness  $T_p$ . All dimensions are summarized in Table.1.

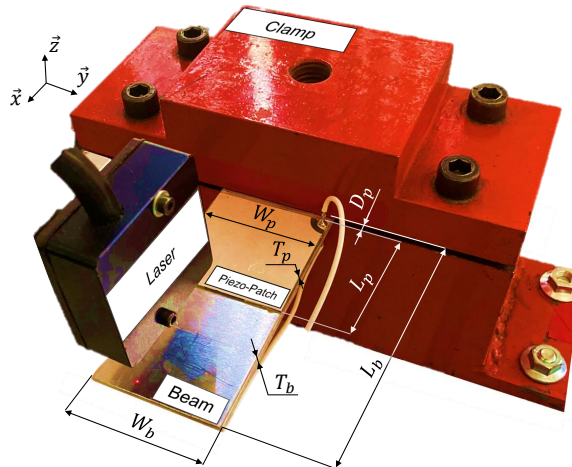


FIG. 4.—: Experimental beam with reference dimensions

TABLE 1: Experimental setup Dimensions

	Dimensions			Materials
	Name	Value	Unit	
Beam	$L_b$	150	[mm]	Aluminium
	$W_b$	52	[mm]	
	$T_b$	3	[mm]	
	$D_p$	3	[mm]	
Piezoelectric Patch	$L_p$	50	[mm]	Pz26
	$W_p$	50	[mm]	
	$T_p$	1	[mm]	

One piezoelectric transducer is set as an actuator with  $v$  as the input signal and the other as a sensor monitoring an image of the beam bending curvature  $x_1$ . A laser is used to monitor the displacement of the free beam end  $x_2$ . All devices used to run the experiment are described in Figure 5 and all the hardware information are given in appendix A 8.

#### 3.2. Experimental Model

The DRL algorithm has to be trained by multiple interactions with the environment to be confident in the parameter tuning of the control law according to the Reward definition. An episode run on the experimental setup is achieved in 1 minute, while it takes only 1 second with the model. Furthermore, tuning parameters that lead to instabilities do not need to be tested on the real structure.

The experimental model is built from the measurements. A band-limited white noise  $v(t)$  defined in Equation 4 is used as an input for the identification process and is applied to the actuator transducer with a noise power equal to 0.1, a sampling frequency  $F_s = 10$  kHz, and a second order filter considering a cut-off frequency  $F_c = 1000$  Hz.

The measured signals  $x_1$  and  $x_2$  allow the identification of two Single Input Single Output (SISO) systems  $G_1$  and  $G_2$  defined by the following transfer functions:

$$\begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} = \begin{bmatrix} G_1(s) \\ G_2(s) \end{bmatrix} V(s) \quad (14)$$

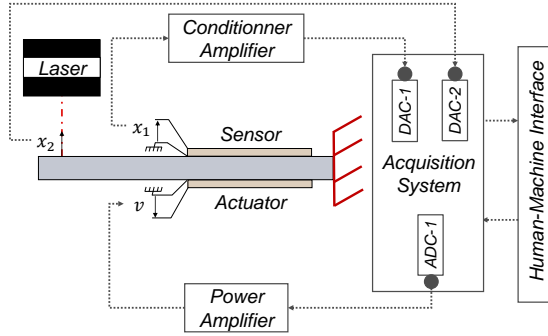
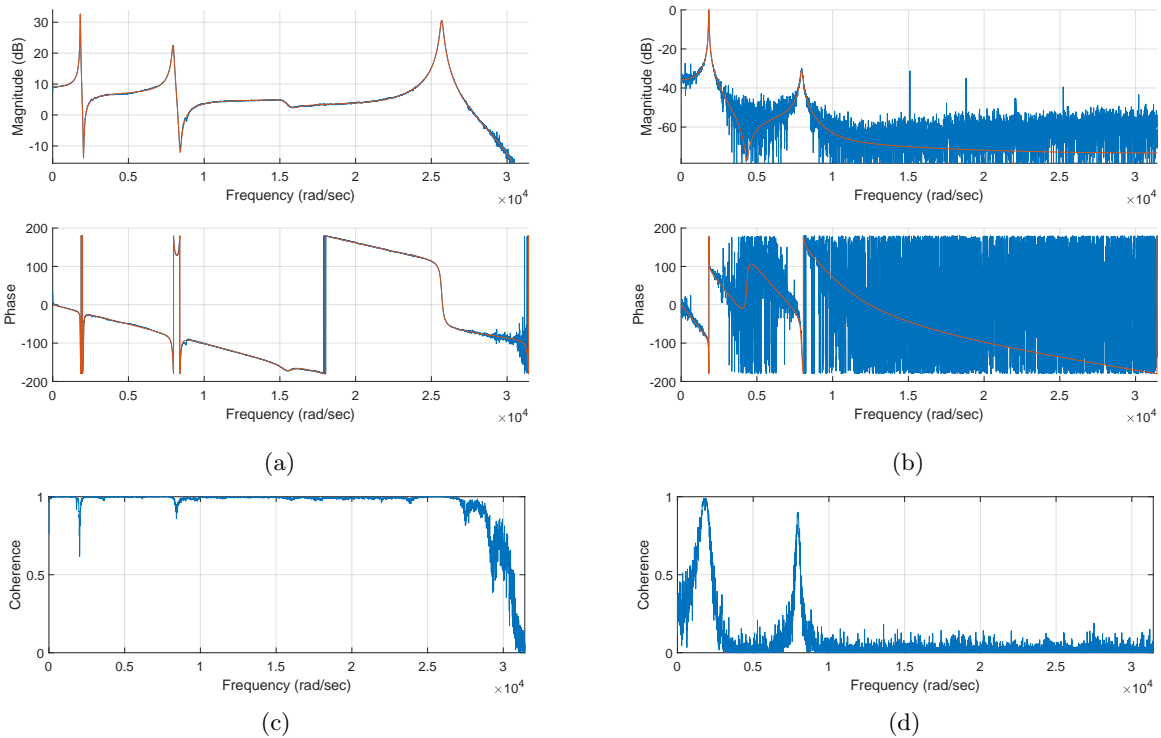


FIG. 5.—: Experimental setup: Scheme

FIG. 6.—: (blue line) Bode diagram with transfert function estimated from measurements and (red line) the identified model between 0-31400 [rad/sec]: (6a)  $G_1$  and (6b)  $G_2$  with (6c-6d) their corresponding spectral coherence

With  $X_1(s)$ ,  $X_2(s)$  and  $V(s)$  the Laplace transform of  $x_1$ ,  $x_2$  and  $v$  respectively.

To build the experimental model from the measured frequency response functions, poles and zeros estimation is chosen, such as:

$$G_{i=\{1;2\}}(s) = \frac{\kappa \prod_{i_z=1}^{n_z} (1 - z_{i_z}^{-1} s)}{s^{n_0} \prod_{i_p=1}^{n_p} (1 - p_{i_p}^{-1} s)} \quad (15)$$

with  $\kappa \in \mathbb{R}$  a constant gain,  $n_0 \in \mathbb{Z}$  is the number of null poles (integrator if  $n_0 > 0$ , derivative if  $n_0 < 0$ ),  $z_i \in \mathbb{C}$  zeros and  $p_i \in \mathbb{C}$  poles of the system,  $n_z \in \mathbb{N}$  the number of zeros and  $n_p \in \mathbb{N}$  the number of poles,  $G_i$  must be proper ( $n_p + n_o \geq n_z$ ).

The numerical transfer function for  $G_1$  includes 12 poles and 10 zeros, while that for  $G_2$  includes 7 poles and 6 zeros, covering a frequency range of 10 to 5000 Hz. It is important to consider the spillover effect for higher frequencies, even if the input disturbance is below 1000 Hz. The transfer functions obtained from experimental measurements and the identified model are shown in Figures 6a and 6b. Within this frequency range, the piezoelectric sensor can observe four modes, but the laser can only detect two.

The spectral coherence between the two piezoelectric transducers, as shown in Figure 6c, is close to one within the desired frequency bandwidth. Between the piezoelectric actuator and the laser, the spectral coherence shown in Figure



6d is high only at frequencies close to the natural frequencies of the beam. A pair of zeros is added at 660 Hz to fit the experimental case even if it cannot be clearly identified on the measured frequency response function due to noise and laser sensitivity. To reduce this noise, the input power of the disturbance has been increased until saturation due to the acquisition system voltage limitations.

To ensure the model fits the experimental setup, the output signals of the model and the experiment need to be as close as possible. Figure 7 displays the time signals from the sensors in the case of band-limited white noise disturbance, with identical inputs for both the experiment and the identified system.

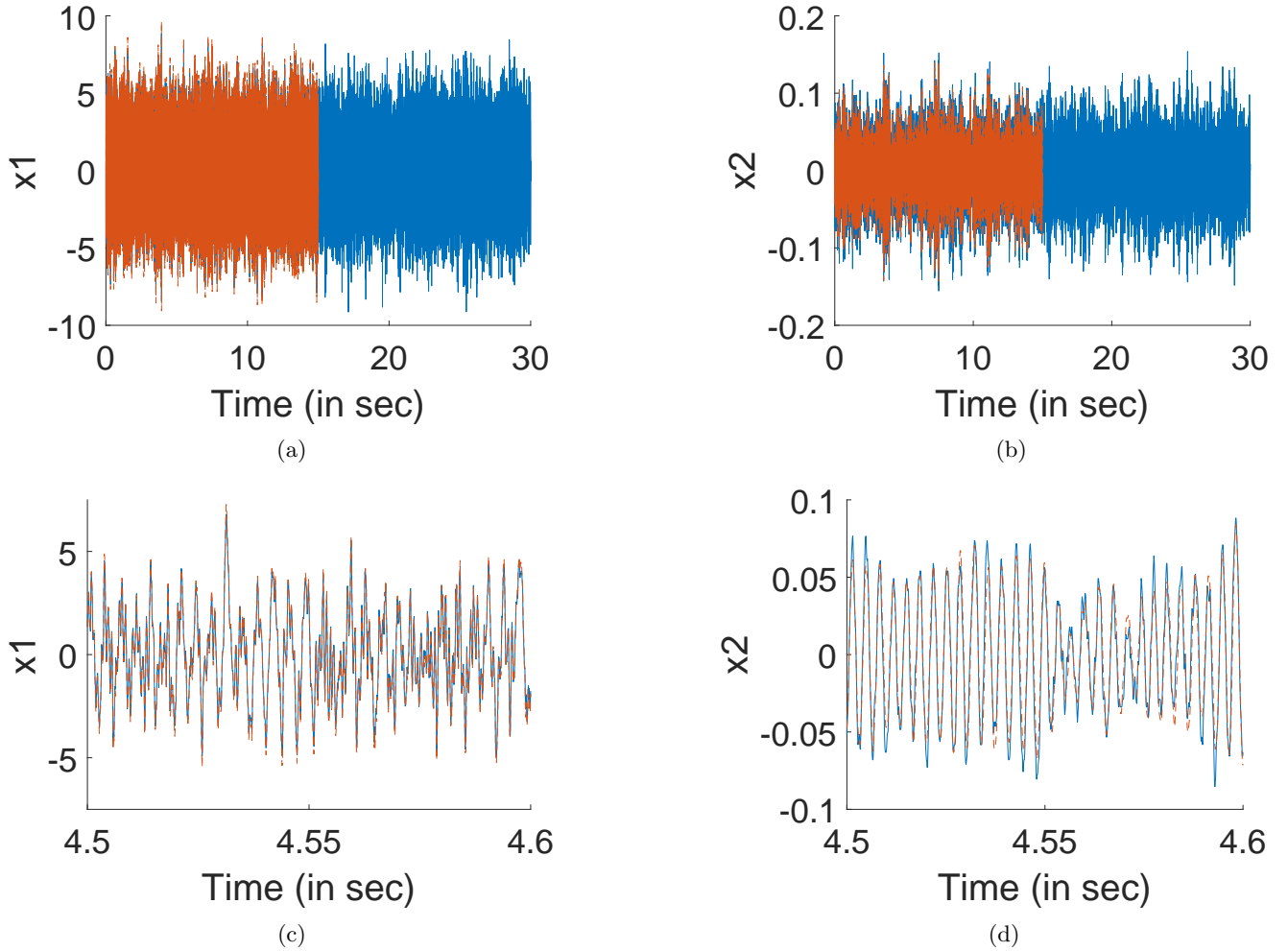


FIG. 7.—: Time measurements (blue line) from the experimental setup and (red dashed line) identified model during 25 sec and between seconds 4.5 and 4.6: piezoelectric (7a-7c) and laser (7b-7d) sensors

The normalized RMS error between the model output and the experimental structure is 0.11 for  $x_1$  and 0.28 for  $x_2$ . This difference is due to measurements and transfer function estimations but it is negligible for the purpose of this work. Therefore, it can be concluded that the model response tendency fits the experimental behavior.

### 3.3. Reference control law tuning

The control objective is to minimize the free-beam end vibrations  $x_2$  with two approaches, considering band limited noise  $v_{d1}$  ( $F_s = 10000Hz$  and  $F_c = 1000Hz$ ) or chirp  $v_{d2}$  ( $f_0 = 10Hz$  to  $f_1 = 5000Hz$   $t_f = 30sec$ ) disturbance in order to compared with  $H_2$  and  $H_\infty$  norm minimization. The feedback control is implemented between the two piezoelectric elements of the beam.

To improve stability and damping on the structure, the chosen control law  $C(s)$  is a first order pseudo lead-lag compensator with three parameters to tune, such as:

$$C(s) = K_c \frac{s - z_0}{s - p_0} \quad (16)$$

with  $K_c \in \mathbb{R}$  the gain,  $z_0 \in \mathbb{R}$  and  $p_0 \in \mathbb{R}$  the pseudo lead-lag values. The objective of this control law is to minimize the vibrations at the free-beam end  $x_2$ , by using the pseudo lead compensator to increase the stability and the lag to

reduce the steady state error. To do so, the three parameters of the control law need to be properly tuned.

### 3.4. DRL for control vibration

The presented pseudo lead-lag control law (16) is now tuned using the DRL algorithm and a TRPO optimization method for NN training. To apply the DRL method to a vibration control problem, interactions between the algorithm and environment, driven by Observations, Actions, and Rewards, must be defined.

The environment is defined by the transfer functions  $G_1$  and  $G_2$ , estimated from the experimental setup. The agent only requires a single interaction with the environment before adjusting its policy by updating both the Critic and the Actor neural networks, given the consistent nature of the structural properties that remain unchanged over time and across episodes. The control law applied to the system is also considered part of the environment and its parameters are the actions chosen by the Actor.

Constraining Action boundaries to the Actor helps to decrease the NN training time since it has to search for a value within a bounded domain. The pseudo lead and the lag coefficients boundaries are defined according to the sample frequency  $F_s$  of the signal and the Shannon criterion, such as  $z_0 \in [-F_s\pi; F_s\pi]$  and  $p_0 \in [-F_s\pi; F_s\pi]$ . The controller gain  $K_c$  boundaries are chosen in close proximity to the system's stability region, which can be determined through root locus computation. Stability is ensured when the real part of the poles is negative. As an example, Figure 8 displays the root locus of the closed-loop transfer function model  $G_{c,1}$  between the piezoelectric actuator and sensor with a pseudo lead-lag controller, where  $z_0 = 0$  and  $p_0 = 6280$ , based on the identified poles and zeros. To compare with the experiment, poles and zeros are identified from measurements using the same controller and  $K_c$  variations from  $K_{c_{min}} = -0.04$  to  $K_{c_{max}} = 0.5$ .

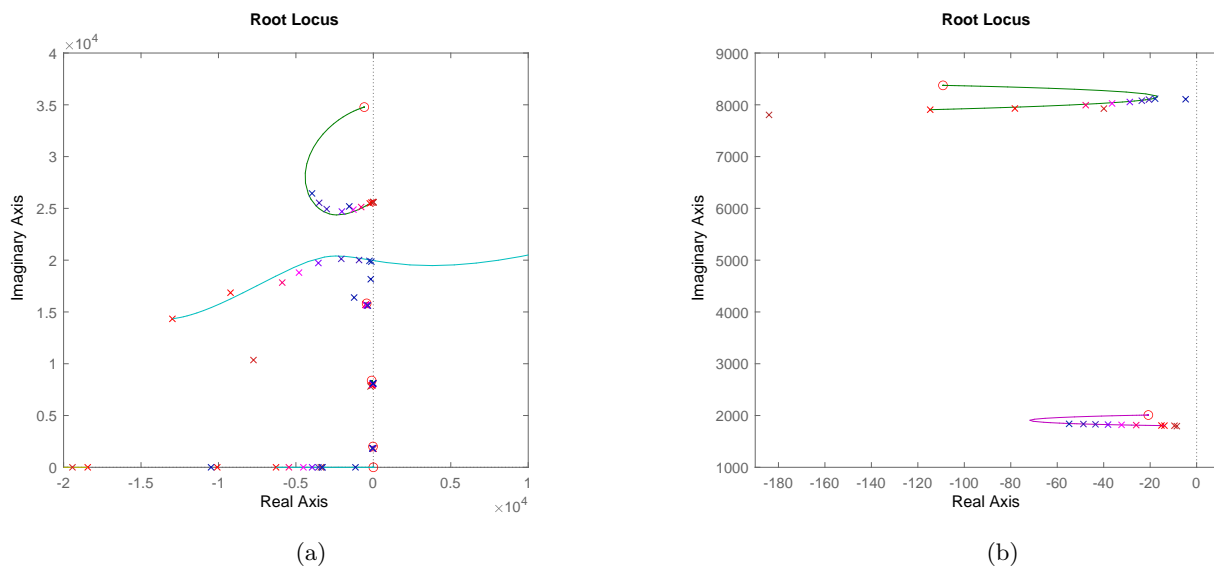


FIG. 8.—: Root locus from identified controlled model  $G_1C(s)$  open loop with (8a)  $z_0 = 0$  and  $p_0 = 6280$  and (8b) a zoom around the two first poles and ( $\times$ ) poles estimated from measurement on the experimental setup when control is on for different values of  $K_c$  from ( $\times$  dark red)  $K_{c_{min}} = -0.04$  to ( $\times$  dark blue)  $K_{c_{max}} = 0.5$

With these  $z_0$  and  $p_0$  pseudo lead-lag parameters, stability is guaranteed for  $K_c < 0.6$ . However, experimental measurements show that the system becomes unstable when  $K_c > 0.5$ . The model overestimates the range of  $K_c$  values that lead to a stable system. Additionally, it does not consider instability coming from the second pole and creates a fictive zero with a negative real part in the high frequencies above the range of interest. Considering different pseudo lead-lag controllers, the system instability occurs for different values of  $K_c$ . In order to include all gains leading to a stable system, the pseudo lead-lag boundaries are set with  $K_c \in [-1; 1]$ .

The sensors on the defined system provide time data observations from the environment. In the context of vibration mitigation, it is not necessary to forecast time data because using each time value of the signal will not accurately represent the system's overall behavior. Optimization of the system will not occur at each time step. Alternatively, significant metrics can be employed to define environmental observations, such as the root mean square (RMS) of the system's output values when the disturbance signal is  $v_{d1}$ , such as:

$$\begin{bmatrix} S_1 \\ S_2 \end{bmatrix}_{v_{d1}} = \begin{bmatrix} rms(x_1) \\ rms(x_2) \end{bmatrix} \quad (17)$$

Considering the chirp disturbance signal  $v_{d2}$  as input, the maximum of the system's output time values are used as Observations:

$$\begin{bmatrix} S_1 \\ S_2 \end{bmatrix}_{v_{d2}} = \begin{bmatrix} \max(|x_1|) \\ \max(|x_2|) \end{bmatrix} \quad (18)$$

These metrics allow to reduce the amount of data input into the agent without losing the meaning of the information on state of the vibrating system. They correspond to  $H_2$  and  $H_\infty$  norms.

The objective of vibration mitigation is given to the DRL algorithm through a definition of the Reward function. DRL algorithm adapts its policy in order to increase a numerical Reward which is computed based on the environment behavior after applying the chosen control parameters. For vibration minimization, rewards are defined according to the two cases studied. Considering the input disturbance  $v_{d1}$ , reward  $R_1 \in \mathbb{R}$  inspired by the  $H_2$  norm definition in the time domain is defined such as:

$$R_1 = 20 \times \log_{10} \left( \frac{\text{rms}(x_{2_{off}}(t))}{\text{rms}(x_{2_{on}}(t))} + \frac{\text{rms}(x_{1_{off}}(t))}{\text{rms}(x_{1_{on}}(t))} \right) \quad (19)$$

with  $[x_{1_{off}}; x_{2_{off}}]$  output time signals when the system is not controlled and  $[x_{1_{on}}; x_{2_{on}}]$  output time signals when the system is controlled.

Based on  $H_\infty$  norm definition in the time domain, with the chirp input disturbance  $v_{d2}$ , Reward  $R_2 \in \mathbb{R}$  is defined as:

$$R_2 = 20 \times \log_{10} \left( \frac{\max(|x_{2_{off}}(t)|)}{\max(|x_{2_{on}}(t)|)} + \frac{\max(|x_{1_{off}}(t)|)}{\max(|x_{1_{on}}(t)|)} \right) \quad (20)$$

To fit the experimental setup, two conditions are added to the previous Rewards. The first condition constrains the piezoelectric sensor to an output voltage between  $-10V$  and  $10V$  according to the constraint of the acquisition system. The second condition is a restriction on the gain margin which must be less than  $5dB$ . For all of these cases, Rewards are set to  $R_{1,2} = -1$  if the aforementioned conditions are not satisfied. Figure 9 shows the variation of  $R_1$  and  $R_2$  with the gain  $K_c$  for different values of  $(z_0, p_0)$  tested on the model and a time signal lasting 30 seconds for  $v_{d1}$  and 60 seconds for  $v_{d2}$ .

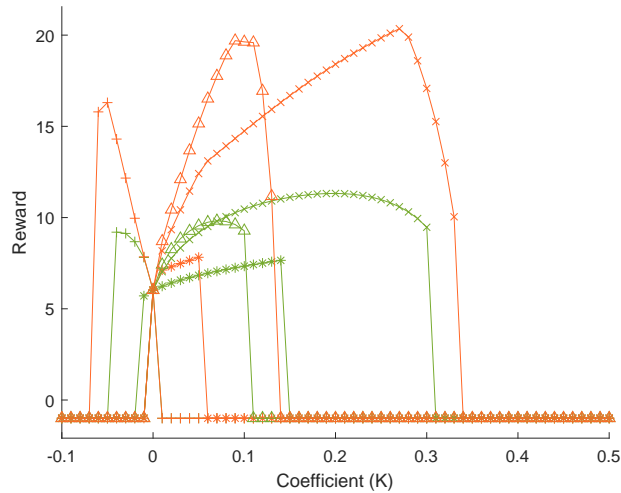


FIG. 9.—: (green line) Reward  $R_1$  and (orange line)  $R_2$  variations with  $K_c$  for different pseudo lead-lag filter  $z_0$  and  $p_0$  combinations : (x)  $z_0 = 6280$ ,  $p_0 = -6280$ ; (+)  $z_0 = -31400$ ,  $p_0 = -6280$ ; ( $\Delta$ )  $z_0 = 31400$ ,  $p_0 = -12560$  and (\*)  $z_0 = -6280$ ,  $p_0 = -21980$  tested on the model

Figure 9 reveals the instability of the system when the reward begins to decrease. As the system approaches instability,  $x_1$  progressively increases with  $K_c$  until it diverges. The Reward definitions introduce knowledge of control instability to the Reinforcement Learning algorithm by using a penalty value. The curves of  $R_2$  are less smooth than those of  $R_1$ . This phenomenon is a result of the reward definition that considers the maximum amplitudes of the time signals and the velocity of the chirps. If the chirp velocity is different, the change in slope will occur at different values of  $K_c$ . The energy is concentrated on each frequency contained in the chirp signal, and the movement toward instability could affect other frequencies that are not excited. This effect is observed when the system is truly unstable.

Now that interactions between environment and algorithm have been defined within the specific frame of vibration mitigation through Observation metrics choice, Action range limitations, and Reward function definitions, the next section presents the final experimental results.

#### 4. RESULTS

In the following section, the control law tuning is performed by training DRL agents with the two defined disturbance signals  $v_{d1}$  and  $v_{d2}$ . Besides, parametric minimization based on  $H_2$  and  $H_\infty$  norms are run as references to tune controllers. Finally, both approaches are compared in terms of controller performances.

##### 4.1. Parametric norm minimization

A parametric approach is performed on the model, running through a range of potential values for the control parameters  $K_c$ ,  $z_0$ , and  $p_0$  within predefined boundaries, with the objective of minimizing the  $H_2$  and  $H_\infty$  norm on  $G_{c,1}$  and  $G_{c,2}$ . Parametric estimation allows to cover all possible solution domains to identify the irregularities and instabilities of the system. The controller parameters tuned and presented in Tables 3 and 4 are used as references for comparison with the DRL tuning method.

##### 4.2. Deep Reinforcement Learning Training

The Critic and Actor NNs dimensions used for this case are specified in Table 2. The dimensions of the NNs have been manually adjusted and will be justified in the discussion section. Appendix B 9 presents all the hyperparameters of the TRPO training.

TABLE 2: Actor and Critic Neural Network definition

		Number of Neurons	
Layer	Name	Actor	Critic
Input	$N_i$	2	2
Hidden Layer 1	$N_{hu}$	25	25
Hidden Layer 2	$N_{hu}$	25	25
Hidden Layer 3	$N_{hu}$	25	25
Hidden Layer 4	$N_{hu}$	25	25
Output	$N_o$	2	1

To ensure consistency of the method, 10 Agents are trained on the model for the two disturbance cases. Each Agent is initialized with random weights and bias for the input and hidden layers. The TRPO NN optimization method requires starting within a trust region when the structure is stable. One known stable solution is the passive system ( $K_c = 0$ ). To guarantee this constraint, the Actor NN outputs are initialized with output layers weights and bias set to zero.

Algorithms run until convergence of all agents is guaranteed to be close to the maximum reward. Figures 10 and 11 show the average reward variation through Episodes. For comparison, two indicative rewards are added to the figures: the reward for the uncontrolled structure and the reward for the structure with the reference controller tuned using parametric norm minimization. The figures show an overall tendency for rewards to increase with episodes, starting from a reward close to the uncontrolled case and moving above the referenced control case. Due to the reward definition, the NN initialization and a part of randomness in the NN training process, the final maximum Reward reached after at the end of the training can vary. Also, the training progress is not exactly the same for all the Actors. Considering the disturbance case with  $v_{d1}$ , downward variations can be observed during the training.

These variations can be explained by two main causes. First, little variations of the control parameters can turn the system from stable to unstable due to the reward definition or proximity to the unstable zone with loss of damping. Second, over-fitting in the training can occur since the maximum reward has already been reached and every other tuning can only lead to lower rewards. This over-fitting hypothesis is justified since the average reward stops increasing after 10000 episodes. Regarding the chirp disturbance  $v_{d2}$ , the average reward tendency is linear after 2000 Episodes and there are fewer downward variations. As shown in Figure 9, the training reaches a maximum expected reward of approximately 20.

Table 3 and 4 show the DRL control law tuning after 20000 and 30000 Episodes respectively. Table 3 shows that all 10 training sessions lead to a reward above the parametric one. However, two of the sessions (Training number 7 and 10) led to a different controller design and a lower reward compared to the others. This phenomenon highlights a limitation of the DRL method as it may only reach a local maximum. This limitation is also evident in table 4 for training number 1. In this case, the reward range at the end of the DRL training is closer to the reward corresponding to the parametric controller applied to the structure. Based on the reward definition, DRL tuning yields better performance on average than parametric norm minimization tuning for the two tested input signal disturbances on the modeled beam. In the next section, the mitigation performance of the tuned controllers will be examined.

TABLE 3: Training Results after 20000 Episodes and control law values and performances on model:  $v_{d1}$  case.

Tuning Method	$R_1$	$R_{V1}$	$ R_1 - R_{V1} $	$K_{c_{opt}}$	$z_{0_{opt}}$	$p_{0_{opt}}$	RMS( $x_1$ )	RMS( $x_2$ )	$\ G_{c,1}\ _2$	$\ G_{c,2}\ _2$
Control off	6.02	-	-	0	0	0	1.9626	0.0306	4.21	0.0346
Parametric										
$\min(\ H\ _2)$	10.59	-	-	0.16	6280	-9425	1.89	0.0130	3.01	0.0142
Training DRL										
1	11.46	11.31	0.15	0.21	5255	-6231	1.52	0.0125	2.63	0.0138
2	11.49	10.96	0.53	0.23	5332	-6257	1.55	0.0123	2.69	0.0135
3	11.41	11.14	0.28	0.27	5004	-5902	1.66	0.0121	2.85	0.0134
4	11.46	10.89	0.57	0.26	5266	-6232	1.61	0.0122	2.78	0.0134
5	11.50	11.34	0.15	0.25	5202	-6118	1.58	0.0122	2.76	0.0134
6	11.50	11.36	0.14	0.23	5269	-6237	1.55	0.0123	2.68	0.0136
7	11.20	11.21	0.01	0.07	7488	-2580	1.62	0.0127	2.91	0.0140
8	11.50	11.49	0.01	0.24	4960	-5852	1.59	0.0121	2.76	0.0135
9	11.51	11.02	0.49	0.23	5106	-6019	1.56	0.0122	2.70	0.0135
10	11.23	11.22	0.01	0.10	6345	-3407	1.63	0.0125	2.83	0.0139
<b>Mean</b>	<b>11.43</b>	<b>11.19</b>	<b>0.23</b>	<b>0.21</b>	<b>5523</b>	<b>-5484</b>	<b>1.59</b>	<b>0.0123</b>	<b>2.76</b>	<b>0.0136</b>
<b>Std</b>	<b>0.11</b>	<b>0.18</b>	<b>0.21</b>	<b>0.06</b>	<b>751</b>	<b>1266</b>	<b>0.04</b>	<b>0.0002</b>	<b>0.08</b>	<b>0.0002</b>

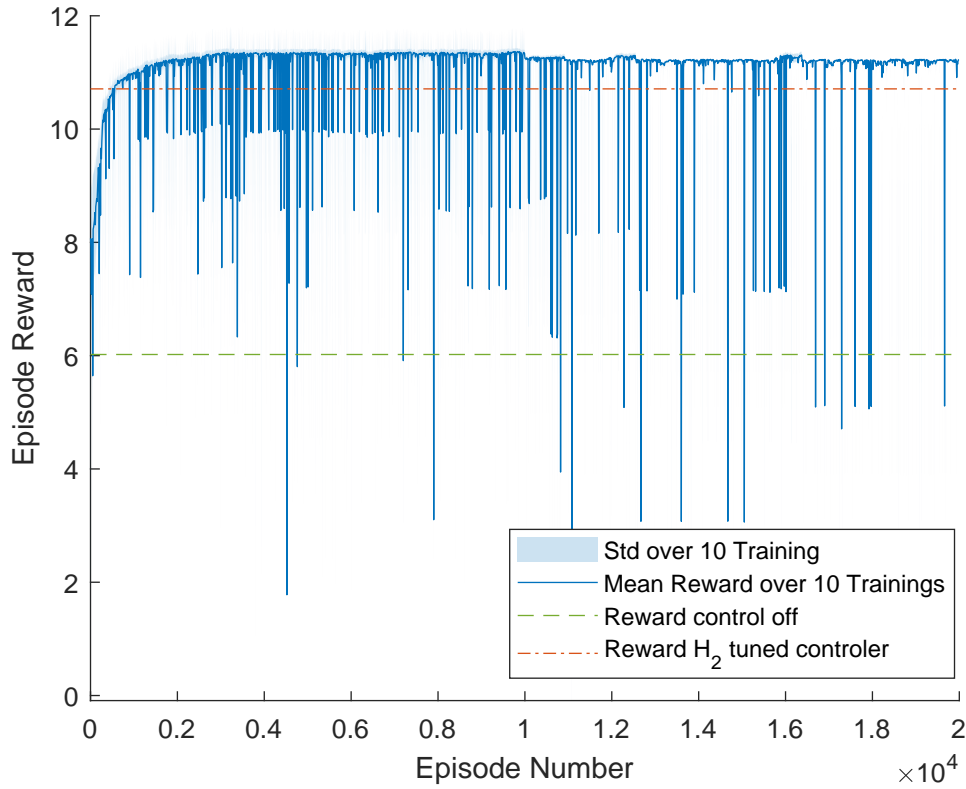
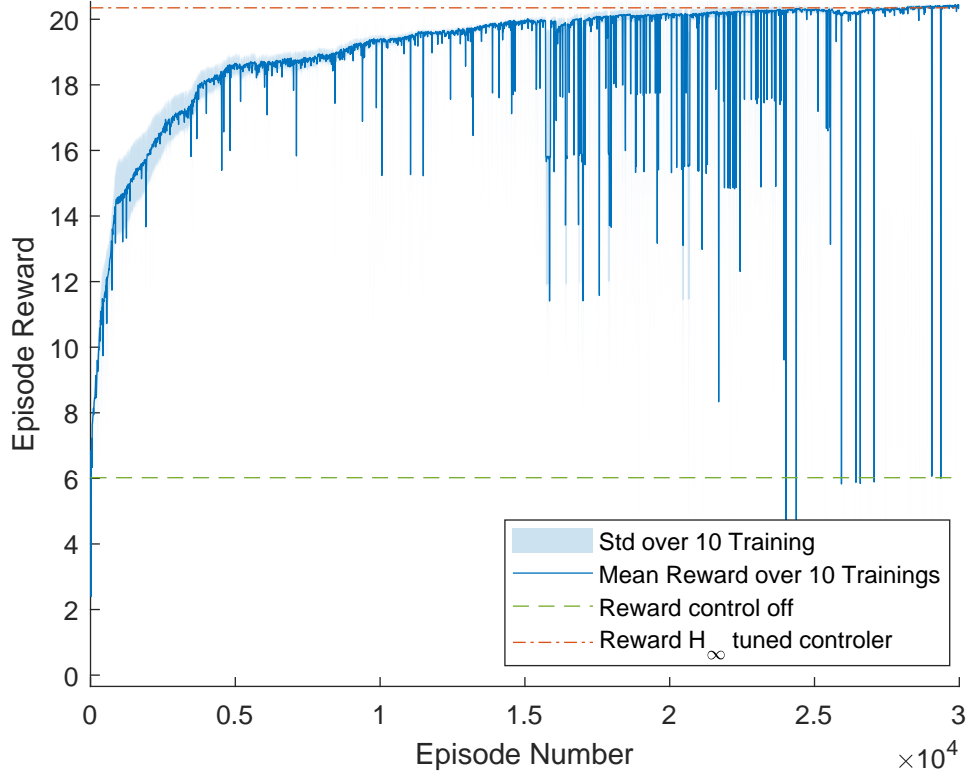


FIG. 10.—: Training on model with disturbance  $v_{d1}$ : (dark blue) Mean Reward over all training, (light blue) the corresponding standard deviation area at each Episode Step, (green - -) the equivalent reward when the control is off and (red .-) the equivalent reward with  $H_2$  norm minimisation controller parameters.

TABLE 4: Training Results after 30000 Episodes and control law values and performances on model:  $v_{d2}$  case.

Tuning Method	$R_2$	$R_{V2}$	$ R_1 - R_{V1} $	$K_{c_{opt}}$	$z_{0_{opt}}$	$p_{0_{opt}}$	MAX( $x_1$ )	MAX( $x_2$ )	$\ G_{c,1}\ _\infty$	$\ G_{c,2}\ _\infty$
Control off	6.02	-	-	0	0	0	6.7258	0.1193	43.10	0.999
Parametric										
$\min(\ H\ _\infty)$	20.35	-	-	0.270	6280	-6280	2.5796	0.0153	16.46	0.0757
Training DRL										
1	20.34	20.23	0.11	0.18	11602	-9422	1.54	0.020	7.72	0.096
2	20.50	20.41	0.08	0.19	10263	-9663	1.39	0.021	6.97	0.101
3	20.51	19.58	0.92	0.20	9997	-9579	1.41	0.020	7.08	0.101
4	20.49	18.94	1.55	0.20	10159	-9608	1.42	0.020	7.12	0.101
5	20.50	19.35	1.14	0.20	10213	-9684	1.40	0.021	6.99	0.103
6	20.50	20.46	0.04	0.20	10049	-9498	1.42	0.020	7.13	0.101
7	20.50	19.38	1.12	0.20	10003	-9592	1.44	0.020	7.19	0.100
8	20.49	20.51	0.02	0.20	10291	-9682	1.40	0.021	7.02	0.103
9	20.40	19.28	1.12	0.20	10809	-9802	1.58	0.019	7.89	0.094
10	20.39	19.27	1.12	0.20	10709	-9702	1.57	0.020	7.67	0.092
<b>Mean</b>	<b>20.46</b>	<b>19.74</b>	<b>0.72</b>	<b>0.20</b>	<b>10420</b>	<b>-9633</b>	<b>1.46</b>	<b>0.020</b>	<b>7.30</b>	<b>0.099</b>
<b>Std</b>	<b>0.06</b>	<b>0.56</b>	<b>0.56</b>	<b>0.007</b>	<b>484</b>	<b>115</b>	<b>0.07</b>	<b>0.0005</b>	<b>0.36</b>	<b>0.003</b>

FIG. 11.—: Training on model with chirp disturbance  $v_{d2}$ : (dark blue) Mean Reward over all training, (light blue) the corresponding standard deviation area at each Episode Step, (green - -) the equivalent reward when the control is off and (red .-) the equivalent reward with  $H_\infty$  norm minimisation controller parameters.

### 4.3. Control performances

The model and experimental structure were subjected to previously tuned controllers to assess their effectiveness in reducing vibration, using significant metrics. An indicative reward was computed from the measured data. The performances of each DRL-tuned controller were compared with the control off and the parametric-tuned controller in Tables 5 and 6. performance of the controller on the model is presented in Tables 3 and 4.

The DRL and parametric methods give different controller tunings, both of which significantly reduce vibration. Mean values are given to highlight the consistency in terms of control performance given by the DRL tuning method, proving its reliability and consistency.

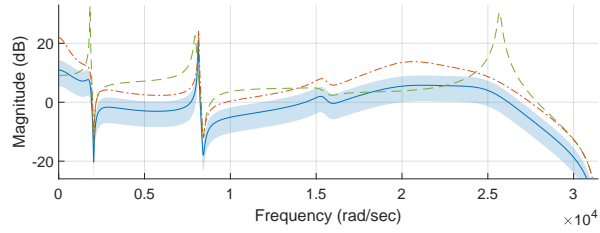
Figures 12 and 13 show the effect of the controllers on the frequency response of the modeled and experimental system. These figures show differences between the modeled behavior of the controller and the experimental response of the system, especially at high frequencies.

Considering the  $v_{d1}$  disturbance for all DRL tuned controllers, the vibration damping performances in the time domain are very close as observed for  $rms(x_1)$  and  $rms(x_2)$ . They give better damping than parametric tuning for both model and experiment. Within the frequency domain, the differences are more visible. All natural frequencies of the system are damped with all tuned controllers. The difference in controller tuning may be due to the parametric definition of the set of sampled values used for norm minimization. Therefore, multiple controllers lead to close vibration reduction performances and DRL tuned controllers are more effective than the parametric controller for both model and experimental.

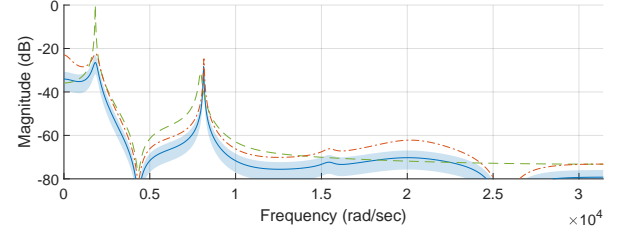
For a chirp  $v_{d2}$  disturbance, the DRL tuned controller gives better attenuation than the parametric tuned controller on the model but not on the experiment. In fact,  $max(x_1)$  and  $max(x_2)$  are lower on the model for the DRL tuned controller compared to the parametric tuned controller, and the contrary occurs on the experimental measurements. These differences are explained by the model underestimating the effect of high frequencies and the static gain of the experimental structure. Also, the reward at the end of DRL training is close to the parametric tuned controller. This explains why the damping performances of the two tuned controllers are close. Nevertheless, the DRL-tuned controllers show a significant reduction in structural vibrations. The tuned controllers for Case 1 and Case 2 are not the same. This proves the efficiency of the method to tune a controller adapted to different input disturbances.

TABLE 5: Case 1: Pseudo Lead-Lag Control Law values and performances measured on the experimental setup

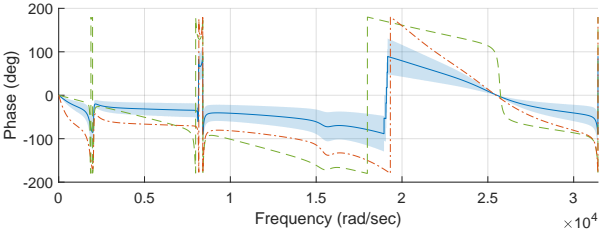
Tuning Method	Tuned Values			Time Metric		Norm		Experimental Reward
Case 1: Noise $v_{d1}$	$K_c$	$z_0$	$p_0$	RMS( $x_1$ )	RMS( $x_2$ )	$\ G_{c,1}\ _2$	$\ G_{c,2}\ _2$	R1
None	0	-	-	1.9400	0.0393	874.3132	8.7597	6.0206
$\ H\ _2$	0.160	6280	-9425	1.6436	0.0234	568.4795	4.5092	9.1268
DRL 1	0.21	5255	-6231	1.38	0.017	540	3.59	11.33
DRL 2	0.23	5332	-6257	1.42	0.017	554	3.49	11.22
DRL 3	0.27	5004	-5902	1.48	0.017	579	3.32	11.25
DRL 4	0.26	5266	-6232	1.48	0.017	584	3.35	11.15
DRL 5	0.25	5202	-6118	1.45	0.017	569	3.36	11.14
DRL 6	0.23	5269	-6237	1.42	0.021	553	3.48	10.19
DRL 7	0.07	7488	-2580	1.46	0.018	587	3.41	10.98
DRL 8	0.24	4960	-5852	1.41	0.018	552	3.37	11.11
DRL 9	0.23	5106	-6019	1.41	0.019	549	3.42	10.86
DRL 10	0.10	6345	-3407	1.42	0.019	556	3.46	10.87
<b>DRL Mean</b>	<b>0.21</b>	<b>5523</b>	<b>-5484</b>	<b>1.43</b>	<b>0.018</b>	<b>562</b>	<b>3.43</b>	<b>11.01</b>
<b>DRL Std</b>	<b>0.06</b>	<b>751</b>	<b>1266</b>	<b>0.03</b>	<b>0.001</b>	<b>16</b>	<b>0.08</b>	<b>0.33</b>



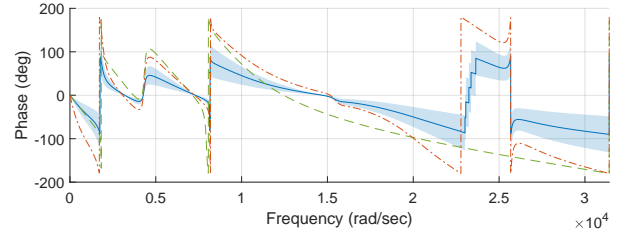
(a)



(b)



(c)



(d)

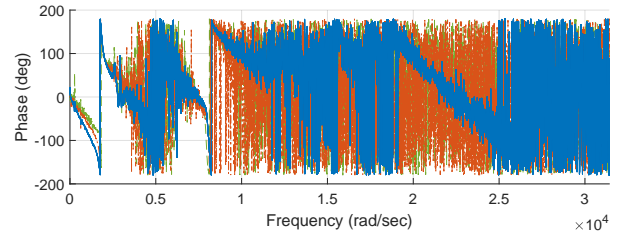
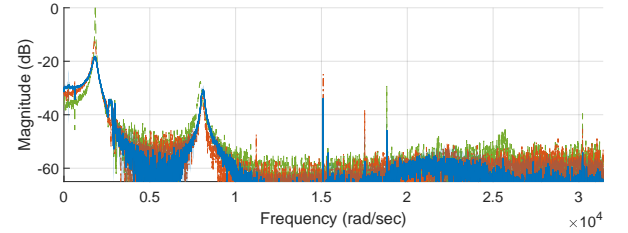
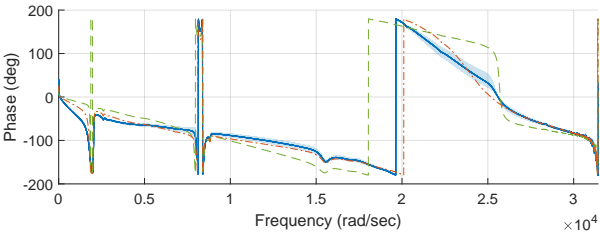
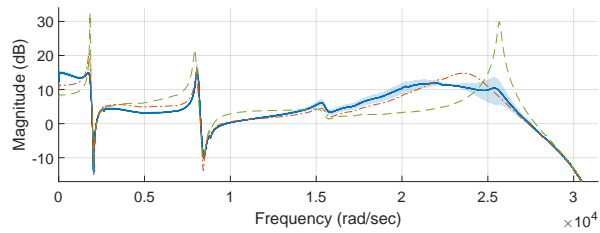
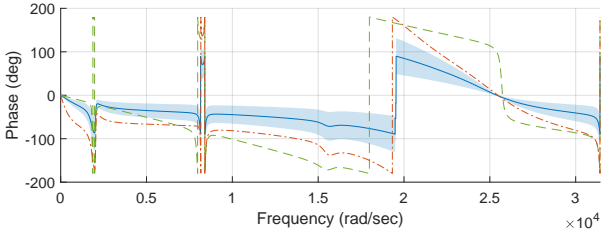
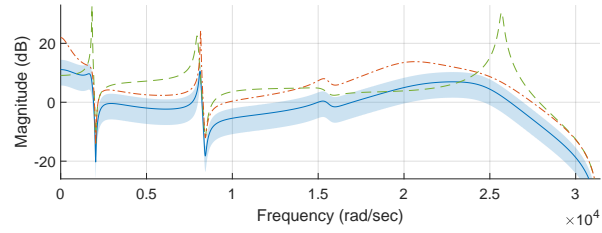


FIG. 12.—: Case  $v_{d1}$ : (green - -) system passive response (red .-) system controlled response with parametric  $H_2$  norm minimisation and (dark blue) mean and (light blue) standard deviation over all responses of the system controlled response with DRL tuned controller: model Fig.12a and experiment Fig.12c  $G_{c,1}$  bode diagram and model Fig.12b and experiment Fig.12d  $G_{c,2}$  bode diagram.

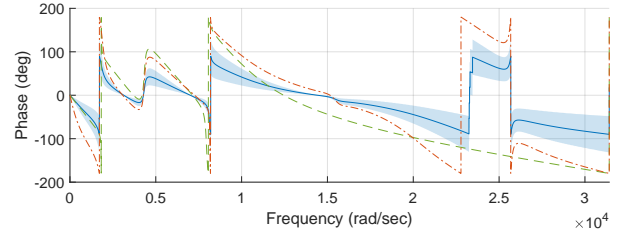
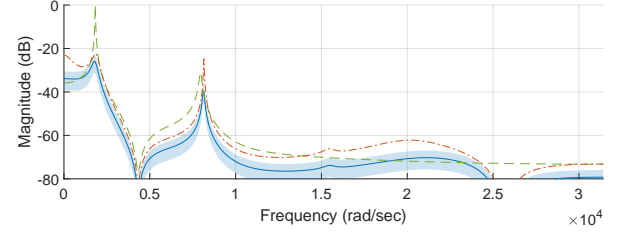


TABLE 6: Case 2: Pseudo Lead-Lag Control Law values and performances measured on the experimental setup

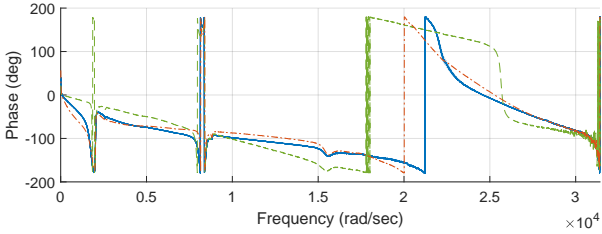
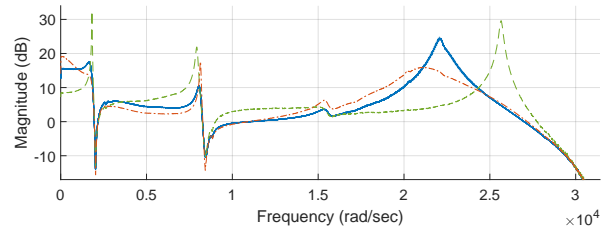
Tuning Method	Tuned Values			Time Metric		Norm		Experimental Reward
Case 2: Chirp $v_{d2}$	$K_c$	$z_0$	$p_0$	$\text{MAX}(x_1)$	$\text{MAX}(x_2)$	$\ G_{c,1}\ _\infty$	$\ G_{c,2}\ _\infty$	$R_2$
None	0	-	-	7.25	0.067	42.42	1.13	6.02
$\ H\ _{inf}$	0.27	6280	-6280	2.23	0.043	9.10	0.10	13.62
DRL 1	0.18	11602	-9422	3.30	0.043	11.46	0.14	11.49
DRL 2	0.19	10263	-9663	3.74	0.057	13.11	0.14	9.88
DRL 3	0.20	9997	-9579	4.41	0.040	15.62	0.14	10.43
DRL 4	0.20	10159	-9608	4.67	0.047	16.71	0.13	9.50
DRL 5	0.20	10213	-9684	5.01	0.058	18.06	0.13	8.27
DRL 6	0.20	10049	-9498	4.25	0.069	14.98	0.14	8.57
DRL 7	0.20	10003	-9592	4.47	0.065	15.86	0.13	8.46
DRL 8	0.20	10291	-9682	5.12	0.062	18.62	0.13	7.95
DRL 9	0.20	10809	-9802	5.94	0.053	24.25	0.13	7.94
DRL 10	0.20	10709	-9702	5.54	0.057	21.07	0.13	7.93
<b>DRL Mean</b>	<b>0.20</b>	<b>10420</b>	<b>-9633</b>	<b>4.65</b>	<b>0.055</b>	<b>16.97</b>	<b>0.13</b>	<b>9.04</b>
<b>DRL Sdt</b>	<b>0.01</b>	<b>484</b>	<b>115</b>	<b>0.80</b>	<b>0.01</b>	<b>3.74</b>	<b>0.01</b>	<b>1.23</b>



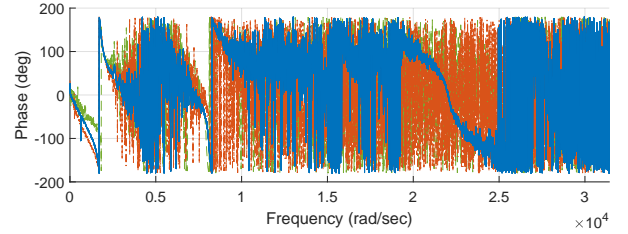
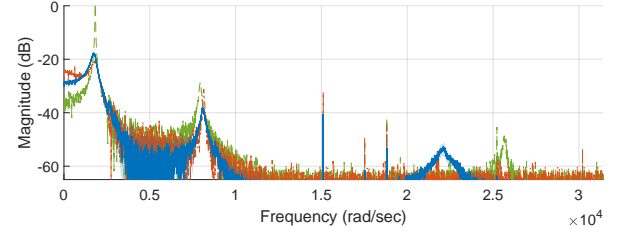
(a)



(b)



(c)



(d)

FIG. 13.—: Case  $v_{d2}$ : (green - -) system passive response (red .-) system controlled response with parametric  $H_\infty$  norm minimisation and (dark blue) mean and (light blue) standard deviation over all responses of the system controlled response with DRL tuned controller: model Fig.13a and experiment Fig.13c  $G_{c,1}$  bode diagram and model Fig.13b and experiment Fig.13d  $G_{c,2}$  bode diagram.

## 5. DISCUSSION

Using a Deep Reinforcement Learning algorithm to tune a control law requires expertise in both Machine Learning and vibration control domains. This section discusses the implementation, usage, and performance of the method for vibration mitigation.

Reinforcement Learning algorithms require a lot of computations with a trial-error process to train Neural Networks. For experimental vibration control applications, these multiple tests are time-consuming considering the acquisition time and the NNs updating process. Moreover, poorly designed and unmastered neural networks can result in closed-loop instability and electrical issues, requiring human intervention to shut down the system. To ensure the efficiency of a DRL-tuned controller, it is more convenient to use an experimental model for running multiple trainings on a powerful hardware computation unit. The model must be as close as possible to the experiment.

The dimensions of Agent Neural Networks must be set according to the problem being solved. Overestimating the NNs dimension can result in null weights and biases, redundant neurons, and increased computation time to update all parameters. Underestimating the dimensions of the neural networks can cause them to struggle to learn properly due to limited memory capacity, increasing the risk of never converging. Researchers are attempting to address this issue by studying how neural networks learn through episodes (Jaunet *et al.* (2020)) and how their design impacts this process (Friedland *et al.* (2018)) considering the influence of the activation function, the number of layers, the number of neurons per layer, the input metrics and the training parameters.

Observations, actions, and rewards define the links between the environment and agent. Meaningful metrics should be used to define observations and actions according to the vibration mitigation objective. It is recommended to normalize the data inputs and outputs when the minimum and maximum values are known to prevent neural network divergence. In this experiment on controlling vibrations, we can estimate action limitations based on system stability. This reduces the domain of parameter exploration and increases training speed.

Depending on the initialization of the neural network, divergence during the training process may occur. Applying the same algorithm can result in diverse outcomes after an equivalent number of training iterations. This highlights the impact of randomness on both computation time and achieved results. The randomness of the training process is determined by the numerical seed. It is recommended to train multiple neural networks in case of random initialization and to continue training until the highest reward is achieved. When using TRPO, it is mandatory to initialize the neural network with a stable solution for the closed-loop since the algorithm searches for a solution by expanding the search space around a starting point, which is known as stable.

The hyperparameters of the algorithm govern the training process. For example, increasing the weight of entropy loss will promote exploration and can avoid staying at local minima. The best tuning for these parameters may change according to the environment, but using the default parameters will fit in many cases. For this cantilever beam experiment, only one interaction with the environment per episode is necessary before updating the Agent parameters according to the Reward. The environment definition remains constant throughout the experiment.

At the end of the training, the DRL method will find a solution, but there is no guarantee that it will reach the optimal solution. It can converge to a local minimum because of overfitting or because the agent needs more episodes to learn. Although this is not the case in this experiment, the end of the training process can be defined by a minimum reward value to reach according to a control performance metric. In this context, the objective is not to find the optimal tuning for a controller, but rather to allow for a trade-off between computation time and control performance. Utilizing DRL to tune a controller enables the creation of an efficient and automatically tuned controller.

The DRL relies on reward definitions to fine-tune the controller. The tuned controllers are not identical because of the reward and the input disturbance signal definition.  $H_\infty$  norm is more robust by considering all frequencies minimization. Since the maximum magnitude frequency changes according to the control parameters, the tuned controller has larger stability margins than the  $H_2$  norm based method. For both disturbances applied to the structure, the tuned control laws present zeros with positive real parts but do not lead to closed-loop instabilities. The use of zeros in the closed-loop transfer function ensures a phase close to 90 at the system's first natural frequency. The tuned controllers tend to soften the system at low frequencies by decreasing the static stiffness. This critical point must be considered to avoid system instability.

## 6. CONCLUSION

As a first step, the DRL approach successfully tunes a three-parameter control law for vibration reduction on an experimental smart cantilever beam. Two input disturbance signals, namely noise  $v_{d1}$  and chirp  $v_{d2}$  are used to train the DRL algorithm, leading to different controllers. The DRL tuned controllers are adapted to the input disturbance signals and ensure efficient vibration reduction. When compared to a parametric controller tuning approach that uses norm minimization, the DRL method's efficiency is proven in both disturbance cases.

The training process for DRL can be time-consuming and computationally expensive, depending on the system's complexity. DRL guarantees to find a solution, but it may only reach a local maximum. The definitions of the DRL elements (environment, observation, reward and action) must be properly defined according to the mitigation objective. In this article, the environment consists of the smart cantilever beam controlled by a pseudo lead-lag controller whose parameters define the DRL actions. Observations are defined by the RMS or maximum temporal response of the structure in coherence with the input disturbance definition. The reward definition is designed to balance between control performance and structure stability.

The experimental setup was modeled to enable multiple tests to be conducted in a short period of time and to fine-tune algorithm and neural network hyper-parameters. By using the model, the DRL algorithm was able to produce the results presented in this article in just a few hours, whereas running the algorithm on the actual experiment would have taken weeks. It is important to note that the model used must closely match the experimental system to ensure efficient tuning of the controller.

Further research is needed to investigate the robustness of DRL to environmental changes. Additionally, DRL should also be tested in more complex systems that include varying states, multiple piezoelectric transducers in a network, and distributive control laws to tune.

## 7. ACKNOWLEDGEMENT

This work was supported by the LABEX CeLyA (ANR-10-LABX-0060) of Université de Lyon, within the program “Investissements d’Avenir” operated by the French National Research Agency (ANR).

## 8. APPENDIX A: EXPERIMENTAL DEVICES REFERENCES

Table 7 references all devices used for this experiment.

TABLE 7: Experimental Devices

Device	Brand	Reference	Sensitivity	Measuring Range
Laser	Micro-Epsilon	LD1607-2	10V/mm	2mm
Acquisition System	DSPACE	DS1104	-	+/-10V
HMI	DSPACE	ControlDesk 7.6	-	-
Conditionner Amplifier	BK	Type 2626	1pC/V	-

## 9. APPENDIX B: EXPERIMENTAL ALGORITHM PARAMETERS

Table 8 resumes the algorithm parameters set during the training. In this study, the agent only interacted once with the environment before changing the policy.

TABLE 8: Algorithm Characteristics

Name	Symbol	Value
Experience Horizon	$N$	1
Mini Batch Size	$M$	1
Entropy Loss Weigth	$w$	0.01
Number of Epoch	$k$	3
Average Estimate Method		”gae”
GAE Factor	$\lambda$	0.95
Conjugate Gradient Damping	$\delta_g$	0.1
KL-Divergence Limit	$\delta$	0.01
Number Iteration Conjugate Gradient	$N_{Cg}$	10
Number Iteration Line Search	$n$	10
Conjugate Gradient Residual Tolerance	$C_{gr}$	1e-8
Normalized Advantage Method		”none”
Advantage Normalizing Window	$N_a$	-
Learning Rate	$L_R$	0.01
Gradient Threshold	$N_{th}$	inf
Gradient Threshold Method		”l2norm”
L2RegularizationFactor	$l2$	1e-4
Training Algorithm		”adam”
Sample Time	$t_s$	1 (Event Base)
Dicount Factor	$\gamma$	0.99

## REFERENCES

- P. Curie, *Oeuvres de Pierre Curie / publ. par les soins de la société française de physique*, edited by Paris (Bibliothèque nationale de France, 1984).
- M.G.Lippmann, *Annales de chimie et de physique: Principe de conservation de l'électricité p.145*, edited by C. P. V. M. M. (Paris) (Bibliothèque nationale de France, 1881).
- J. G. Ziegler and N. B. Nichols, *Transactions of the American Society of Mechanical Engineers* **64**, 759 (2022).
- S. Khot, N. P. Yelve, R. Tomar, S. Desai, and S. Vittal, *Journal of Vibration and Control* **18**, 366 (2011).
- M. M. Jovanović, A. M. Simonović, N. D. Zorić, N. S. Lukić, S. N. Stupar, and S. S. Ilić, *Smart Materials and Structures* **22** (2013), 10.1088/0964-1726/22/11/115038.
- G. Foutsitzi, D. Marinova, E. Hadjigeorgiou, and G. Stavroulakis, in *2003 International Conference Physics and Control. Proceedings*, Vol. 1 (2003) pp. 157–162 vol.1.
- G. Stavroulakis, G. Foutsitzi, E. Hadjigeorgiou, D. Marinova, and C. Baniotopoulos, *Advances in Engineering Software* **36**, 806 (2005).
- J. Tani, J. Qiu, and H. Miura, *Journal of Intelligent Material Systems and Structures* **6**, 380 (1995).
- J. Doyle, K. Glover, P. Khargonekar, and B. Francis, *IEEE Transactions on Automatic Control* **34**, 831 (1989).
- J. Rodriguez, M. Collet, and S. Chesné, *Journal of Vibration and Acoustics* **144** (2022), 10.1115/1.4053358.
- L. Li, G. Song, and J. Ou, *Structural Control and Health Monitoring* (2009), 10.1002/stc.356.
- M. Kwak and D. Sciulli, *Journal of Sound and Vibration* **191**, 15 (1996).
- F. Rosenblatt, *OpenAIRE - Explore* **65**, 386 (1958).
- G.-S. Lee, *Elsevier Science Ltd* **38**, 269 (1996).
- R. Jha and J. Rower, *SMS* **11**, 115 (2002).
- Y. Liang, S. Xu, K. Hong, G. Wang, and T. Zeng, *Measurement and Control* **52**, 1362 (2019).
- S. D.Snyder and N. Tanaka, *IEEE* **6**, 819 (1995).
- C. P. Smyser and K. Chandrashekhara, *Smart Materials and Structures* **6**, 178 (1997).
- Mohit, D. Chhabra, and S. Kumar, *Advances in Aerospace Engineering* **2015**, 1 (2015).
- D. Cardon, J.-P. Cointet, and A. Mazières, *Réseaux n°* **211**, 173 (2018).
- K. J. Lee, in *Hardware Accelerator Systems for Artificial Intelligence and Machine Learning*, Advances in Computers, Vol. 122, edited by S. Kim and G. C. Deka (Elsevier, 2021) pp. 217–245.
- W. Koch, R. Mancuso, R. West, and A. Bestavros, *ACM Transactions on Cyber-Physical Systems* **3**, 1 (2019).
- A. Guerra-Langan, S. Araujo Estrada, and S. Windsor, in *AIAA SCITECH 2022 Forum* (American Institute of Aeronautics and Astronautics, 2022).
- K. P. T. Haughn, C. Harvey, and D. J. Inman, *Communications Engineering* **3** (2024), 10.1038/s44172-024-00201-8.
- Z.-c. Qiu, G.-h. Chen, and X.-m. Zhang, *Aerospace Science and Technology* **118**, 107056 (2021).
- A. Khalatbarisoltani, **26**, e2298 (2019).
- D. Pisanski and L. Jankowski, *Computer-Aided Civil and Infrastructure Engineering* **38**, 1605 (2022).
- J. Panda, M. Chopra, V. Matsagar, and S. Chakraborty, *Computers and Structures* **290**, 107183 (2024).
- L. V. Nguyen, N. T. Nguyen, N. H. Tran, M. Juntti, A. L. Swindlehurst, and D. H. N. Nguyen, *IEEE Wireless Communications* **30**, 174 (2023).
- J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, in *Proceedings of the 32nd International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 37, edited by F. Bach and D. Blei (PMLR, Lille, France, 2015) pp. 1889–1897.
- A. Preumont, “Optimal control,” in *Vibration Control of Active Structures: An Introduction* (Springer International Publishing, Cham, 2018) pp. 259–287.
- M. A. Khushnood, W. Xiaogang, and C. Naigang, *Journal of Vibration and Control* **24**, 1469 (2016).
- M. M. Jovanović, A. M. Simonović, N. D. Zorić, N. S. Lukić, S. N. Stupar, A. S. Petrović, and L. Wei, *FME Transaction* **42**, 329 (2014).
- M. Febvre, J. Rodriguez, S. Chesne, and M. Collet (2023) p. V001T04A001.
- R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (Springer US, Boston, MA, 1992) pp. 1–3.
- K. Fukushima, *IEEE Transactions on Systems Science and Cybernetics* **5**, 322 (1969).
- J. Nocedal and S. J. Wright, eds., “Line search methods,” in *Numerical Optimization* (Springer New York, New York, NY, 1999) pp. 34–63.
- J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, *ArXiv* (2017), <https://arxiv.org/abs/1707.06347>.
- T. Jaunet, R. Vuillemot, and C. Wolf, *Computer Graphics Forum* **39**, 49 (2020).
- G. Friedland, M. M. Krell, and A. Metere, “A practical approach to sizing neural networks,” (2018)