UNIVERSITY OF LEEDS

University of Sheffield

UNIVERSITY of York

Research

Author for correspondence:
Collins O. Ogbodo
e-mail: coogbodo1@sheffield.ac.uk

Adaptive sensor steering strategy using deep reinforcement learning for dynamic data acquisition in digital twins

Collins O. Ogbodo[1], Timothy J. Rogers[1], Mattia Dal Borgo[3] and David J. Wagg[1,2]

[1]Dynamic Research Group, School of Mechanical, Aerospace and Civil Engineering, University of Sheffield, Mapping Street, Sheffield, S1 3JD, UK
[2]The Alan Turing Institute, London, NW1 2DB, UK
[3]Siemens Digital Industries Software NV, 68, Leuven, Interleuvenlaan 3001, Belgium

COO, 0000-0002-3672-0240; TJR, 0000-0002-3433-3247; MDB, 0000-0003-4263-0513; DWJ, 0000-0002-7266-2105

This paper introduces a sensor steering methodology based on deep reinforcement learning (DRL) to enhance the predictive accuracy and decision support capabilities of digital twins by optimizing the data acquisition process. Traditional sensor placement techniques are often constrained by one-off optimization strategies, which limit their applicability for online applications requiring continuous informative data assimilation. The proposed approach addresses this limitation by offering an adaptive framework for sensor placement within the digital twin paradigm. The sensor placement problem is formulated as a Markov decision process (MDP), enabling the training and deployment of an agent capable of dynamically repositioning sensors in response to the evolving conditions of the physical structure as represented by the digital twin. This ensures that the digital twin maintains a highly representative and reliable connection to its physical counterpart. The proposed framework is validated through a series of comprehensive case studies involving a cantilever plate structure subjected to diverse conditions,

THE ROYAL SOCIETY
PUBLISHING

including healthy and damaged conditions. The results demonstrate the capability of the DRL agent to adaptively reposition sensors, improving the quality of data acquisition and hence enhancing the overall accuracy of digital twins.

1. Introduction

The increasing complexity and multidisciplinary nature of engineering systems over the years has continuously motivated the development of new methods for analysis, monitoring and optimization. This increase has led to the emergence of digital technologies such as digital twins—a virtual replica of a physical asset that enables a bidirectional coupling between digital and physical spaces [1]. The digital twin paradigm provides opportunities across different engineering disciplines, including technical risk reduction and reduced cost of experiments in digital engineering, enhanced first-time yield and product optimization in manufacturing, and improved system capability and operational availability in operation engineering [2]. In other fields like precision medicine, digital twins are utilized for patient-personalized medical care [3] and in geophysical science for estimation of ice sheet flow model parameters [4]. Digital twin systems are built from the integration of models and data using sophisticated algorithms, expert know-how and digital connectivity, presenting significant predictive capability when compared to current technologies [5]. Within engineering systems, this integration involves the combination of physics-based and data-driven models for improved reliability of predictive results. Critical aspects such as continual validation and updating ensure that the virtual twin maintains a good representation of the physical system. To achieve this representation, digital twin systems assimilate streams of data from the physical asset in real or near real-time.

While the decreasing cost of sensors and enhanced connectivity has made available a large amount of data, the quality of data assimilated by the digital twin is a key determinant of its predictive accuracy. Given this dependence, data used in the digital twin update process are expected to contain relevant patterns of interest, be characterized by low noise levels and reproducible under the same circumstances [6]. To ensure the virtual representation is tailored to the behaviour of its physical counterpart, a critical part of the virtual-to-physical feedback flow of a digital twin is a dynamic data acquisition process. In scenarios where data are scarce, methods such as active learning and reinforcement learning can guide the collection of additional data that is most relevant to the digital twin's objectives [7].

This work, therefore, presents a novel approach for the dynamic data acquisition process of digital twins using a sensor steering methodology based on deep reinforcement learning (DRL). Our proposed framework learns a sensor steering policy that is deployed online within the virtual space and decides on data acquisition strategies, particularly sensor location, to maximize the information content of acquired data. This approach extends the adaptability of digital twins to the changing behaviours of the physical asset [7].

This paper is organized as follows: in §2, we present a review of previous related methodologies of sensor placement and their respective limitations within the digital twin paradigm. In §3, the problem definition and mathematical abstractions via DRL are provided. In §4, we establish how the proposed methodology integrates into the digital twin paradigm. In §5, we present the results of the proposed strategy when applied to a cantilever plate structure. Finally, in §6, we summarize our main contributions, including limitations and future directions.

2. Background

The problem of optimal sensor placement (OSP) has been well researched across multiple studies within structural dynamics [8,9]. The key aspects of OSP research are the evaluation criteria and optimization algorithms [10]. The evaluation criteria align with specific monitoring objectives and effectively reflect the quality of the sensor placement solution, and the iterative search for
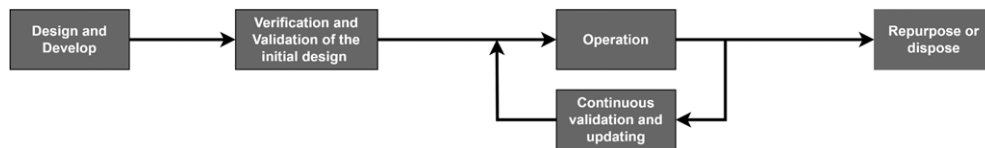
Figure 1. The digital twin life cycle that involves continual updating and validation to maintain accuracy in prediction and decision support.

optimal solutions is largely influenced by the selected criteria's structure, which significantly affects computational efficiency [10]. Different evaluation criteria have been extensively explored, some of which aim to optimize a prespecified objective, such as maximizing a characteristic vibration signal, typically kinetic energy, which has been found to yield suboptimal results owing to its independence on the parameters being estimated [11], modal kinetic energy [12,13] and driving point residues [14,15]. Other criteria are based on maximizing modal identification metrics, like modal assurance criteria [16] and singular value decomposition ratio [17], as well as information-based criteria such as information entropy [18] and mutual information [19]. A summary of these criteria is presented in [10] with their respective merits and demerits. The optimization algorithm, on the other hand, is the computational strategy adopted to either minimize or maximize the selected evaluation criteria, which is usually iterative. A number of optimization algorithms have also been explored. Some of these are evolutionary algorithms, notably genetic algorithms, particle swarm optimization and simulated annealing [9]. Other heuristic algorithms such as the forward and backward sequential sensor placement have also been successfully applied to obtain optimal locations [20]. Recent research has also explored data-driven approaches based on DRL within an optimization formulation to search the placement space for optimal location. DRL was applied in [21] to find the optimal location of two sensors in a base-isolated system with five degrees of freedom (DOFs) using a value-based deep Q-Network agent.

Within the digital twin paradigm, where data assimilation and continuous updating are required, acquiring high-quality data is essential to ensure the digital twin maintains an accurate representation of the physical asset [5]. An essential stage of the digital twin life cycle is continuous validation against operational data, which ensures accuracy in prediction and decision support. This process is highly dependent on the data acquired at every stage of the operational lifespan of the physical asset (figure 1). While existing sensor placement strategies are suitable for traditional structural monitoring operations and focus on finding a single optimal solution that remains fixed throughout the entire lifespan of the structure, the digital twin framework typically requires an adaptive approach. This approach must consider the changing dynamic behaviour of the physical structure and align sensor locations to guarantee the continuous acquisition of high-quality data. This adaptive sensing strategy was researched in [22], which presented an interpretable approach based on an optimal decision tree. The authors proposed a sensor scheduling methodology that was applied on a mathematical abstraction of an unmanned aerial vehicle digital twin based on a model library of damages [23]. However, this strategy assumes that the damage states of the structure are known *ab initio* and, therefore, has a distribution of sensors across the wing of the unmanned aerial vehicle, which results in the deployment of a significant number of sensors.

In this study, we shift from a sensor scheduling approach to a sensor steering approach as shown in figure 2 for the example of a rectangular plate. Sensor steering involves a virtual-to-physical change [2], which offers significant benefits for monitoring structures and further enhancing control. Unlike sensor scheduling, which relies on pre-mounted sensors, sensor steering dynamically adjusts sensor positions based on real-time or near real-time data and evolving conditions of the physical asset. This approach ensures that the digital twin continuously receives the most relevant data. The enhanced methodology provides a robust framework for
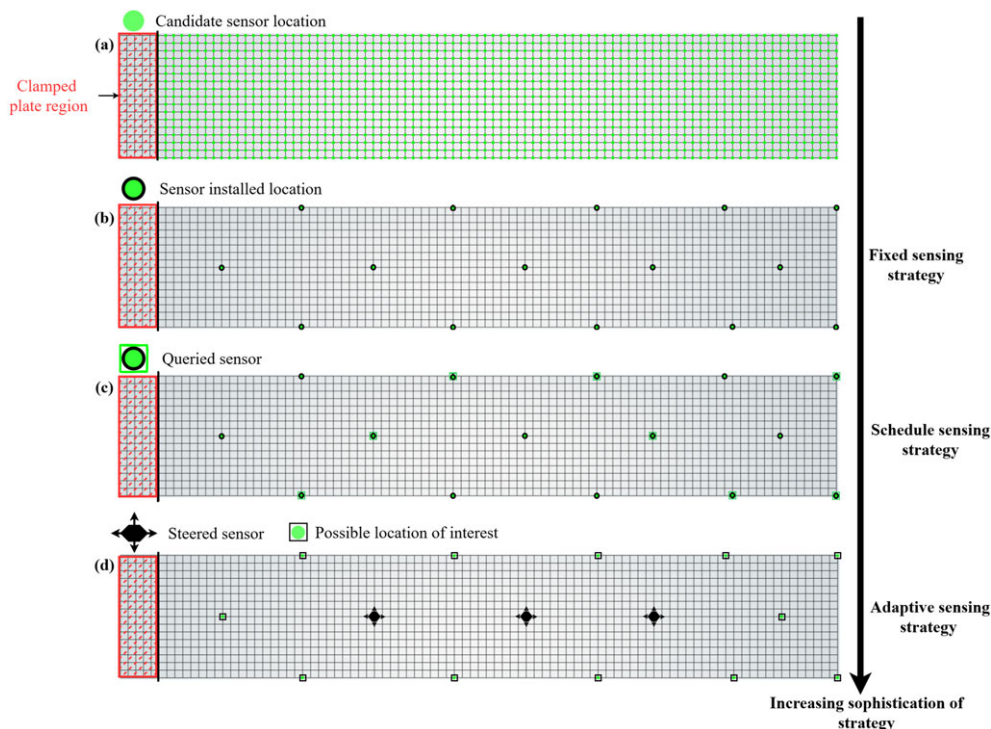
Figure 2. Evolution of sensor placement strategies for the example of a rectangular plate showing (a) a grid of all possible sensor candidate locations, (b) fixed sensing strategy—sensor locations are predefined and remain fixed throughout the structure's lifespan, (c) sensor scheduling strategy—sensor locations are predefined and queried at different times and (d) sensor steering strategy—sensors are adapted to new locations of interest across the lifecycle of the structure.

adaptive sensor placement within the digital twin paradigm, ensuring that the digital twin remains an accurate and reliable representation of the physical asset throughout its operational lifespan.

Our contribution, therefore, differs from previous work in three aspects:

— While previous work presented above is either fixed or scheduling strategies, ours is an adaptive steering strategy that does not involve computing the model's time response for training data.
— Although reinforcement learning (RL) has been explored previously within an optimization setting, we follow a policy-based approach that is suitable for online decision processes such as digital twins and other data-driven models.
— Our strategy employs a distributional RL framework, presenting opportunities for placement decisions based not only on the expected return but also on alternative metrics such as variance and conditional value at risk of the return distribution.

3. Problem formulation

The primary objective of this study is to develop a sensor steering strategy that dynamically adapts the location of sensors on a structure in response to changing conditions. For this, we adopt an agent-based approach where we learn how to act (change the sensor position) given the current state of the physical asset as represented by the digital twin. This section presents a mathematical formulation of the problem using the Markov decision process (MDP) and its implementation with DRL.
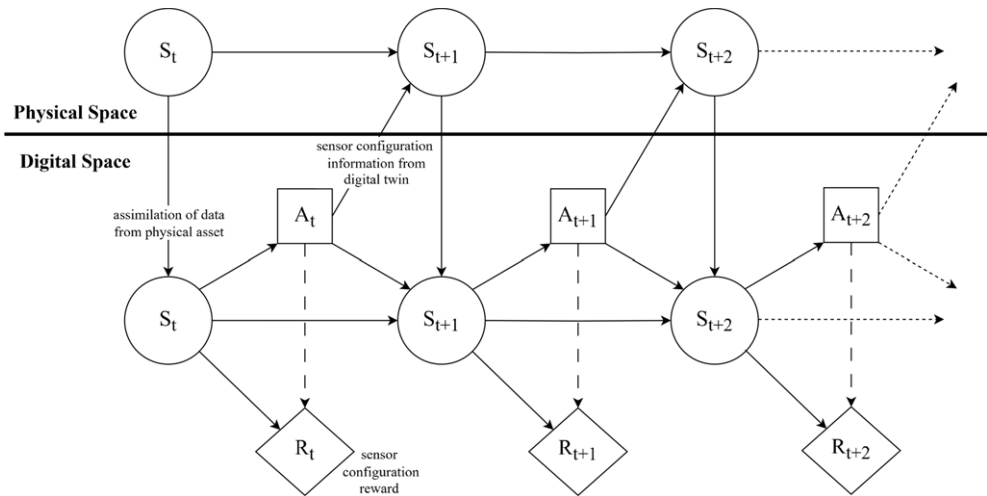
Figure 3. Graphical model of dynamic decision interaction between physical and digital space showing sensor steering information exchange.

(a) Markov decision process formulation

The sensor steering problem can be formulated as a sequential decision-making process, where the goal is to determine the optimal sensor configuration at each time step. This involves deciding which sensors to move and in which direction, based on the current state of the digital twin. At each step, the sensor configuration is dependent only on the selected sensor and the steering direction, which can be modelled with the MDP formulation defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where

— $s \in \mathcal{S}$ is the set of states representing the possible sensor configurations;
— $a \in \mathcal{A}$ is the set of actions containing all possible sensor motion;
— $\mathcal{P} : p(s'|s, a) := Pr\{S_{t+1} = s'|S_t = s, A_t = a\}$ is the state transition probability matrix, representing the dynamics of the virtual environment;
— $\mathcal{R} : R(s, a)$ is the reward function, which quantifies the quality of the sensor configuration; and
— $\gamma \in [0, 1]$ is the discount factor, which determines the importance of the final sensor location rewards.

The problem also naturally satisfies the Markov assumption, denoted as

$$Pr\{S_{t+1} = s'|S_t = s, A_t = a\} = Pr\{S_{t+1} = s'|S_t = s_{t,t-1,\ldots,0}, A_t = a\}, \tag{3.1}$$

where $s_{t,t-1,\ldots,0}$ is the state trajectory history. A graphical model of the process similar to that developed by Tezzele *et al.* [24] is shown in figure 3. The graphical model shows the physical-to-virtual and virtual-to-physical data and information interaction.

(i) Action space

A multidiscrete action space [25] is used in the form of a vector of discrete numbers with length equal to two actions: (i) selected sensor $p \in \mathbf{P}$ bounded by the maximum number of sensors specified in the environment set-up; (ii) steering direction $g \in \mathbf{G}$ bounded by the motion directions ($g_0 =$ left, $g_1 =$ right, $g_2 =$ up and $g_3 =$ down). Actions are sampled from a space of $\mathbb{R}^{\mathbf{P} \times \mathbf{G}}$ and executed in the environment. A third action which encoded the number of steps to take in a specific direction was originally considered, but this significantly increased the dimension of the

action space, which reduced the efficiency of learning and hence was not used in this study. The action space is further reduced to a discrete action to enable the use of a value-based learning algorithm. Therefore, a typical action $a = 1$ is interpreted within the environment as $[0, 1]$ which means move sensor p_0 one step in the right direction g_1.

(ii) State space

The state is defined by a multi-binary space [25] of dimension $\mathbb{B}^{n^*}$ where $\mathbb{B} = \{0, 1\}$ and n^* is the number of mesh nodes in the pre-specified placement region of the discretized model. The space is represented by a vector with length n^*. Nodes represent possible sensor candidate positions and are encoded with ones when instrumented with a sensor and zeros otherwise. Each sensor position combination represents a state of the environment.

(b) Reward function

The sensor steering problem is fundamentally built on an OSP formulation, which is in itself a combinatorial problem that involves the search for u candidate positions that maximize a specified objective function given v possible positions, where $u < v$. The space of possible sensor combinations is then given as ^vC_u, which scales up significantly with the size and complexity of the structure. Unlike OSP, which involves running the underlying optimization once to produce an optimal combination result, the sensor steering process is a learning problem that results in a policy and therefore is more computationally intensive, ideally requiring a fast, easy-to-compute objective function. Given the nature of the learning problem, an information-theoretic objective function satisfies our learning requirement, particularly the Fisher information matrix (FIM) metric.

(i) Information theoretic reward

The learning goal, by extension, is to acquire highly informative data, which makes an information entropy reward function suitable. Information entropy serves as a quantitative measure of probabilistic uncertainty in the estimated model parameter given a data stream and provides a basis for determining sensor layouts that minimize this uncertainty [18].

Following the formulation presented in [20], consider a structural model $M(\theta) \in \mathcal{M}$ parameterized by a vector $\theta \in \mathbb{R}^{N_\theta}$, where $\boldsymbol{\theta}$ characterizes either the dynamic structural behaviour or response and $\mathcal{M}$ is the set of models that constitute the digital twin and, consequently, the RL environment. The structural response measured at N_{st} DOFs on the structure is given as $\mathbf{y} \in \mathbb{R}^{N_{st}}$ and model ($M(\theta)$) response predicted at N_{mod} DOFs is given as $\mathbf{x}(\boldsymbol{\theta}) \in \mathbb{R}^{N_{mod}}$. The relation between $\mathbf{y}$ and $\mathbf{x}(\boldsymbol{\theta})$ satisfies

$$\mathbf{y} = \mathbf{L}\mathbf{x}(\boldsymbol{\theta}) + \mathbf{e}(\boldsymbol{\theta}), \tag{3.2}$$

where $\mathbf{x}(\boldsymbol{\theta})$ represents the displacement, velocity or acceleration response of the model and $\mathbf{e}(\boldsymbol{\theta})$ accounts for prediction errors, including both model and measurement errors. The matrix $\mathbf{L}$ is a binary observation matrix that specifies the DOFs monitored by the sensors. In this context, $\mathbf{L}$ is considered a one-hot encoding of the state vector s, which defines the monitored DOFs associated with the response $\mathbf{x}(\boldsymbol{\theta})$. Based on Bayesian statistics, under a Gaussian likelihood assumption the probability density function of $\boldsymbol{\theta}$, given a set of measured data $\mathbf{D} \supset \{y_k \in \mathbf{y} \mid k = 1, \ldots, n\}$, is expressed as

$$p(\boldsymbol{\theta} \mid \boldsymbol{\Sigma}, \mathbf{D}) = c \frac{1}{(\sqrt{2\pi})^n \sqrt{\det(\boldsymbol{\Sigma})}} \exp\left[-\frac{nN_{st}}{2} J(\boldsymbol{\theta}; \boldsymbol{\Sigma}, \mathbf{D})\right] p(\boldsymbol{\theta}), \tag{3.3}$$

where $p(\boldsymbol{\theta})$ is the prior distribution of $\boldsymbol{\theta}$, c is the marginal proportionality constant and $J(\boldsymbol{\theta}; \mathbf{D}, \boldsymbol{\Sigma})$ quantifies the distance between measured and model-predicted response and is given as

$$J(\boldsymbol{\theta}; \boldsymbol{\Sigma}, \mathbf{D}) = \frac{1}{nN_{st}} \sum_{k=1}^{n} [\mathbf{y}_k - \mathbf{L}\mathbf{x}_k(\boldsymbol{\theta})]^\top \boldsymbol{\Sigma}^{-1} [\mathbf{y}_k - \mathbf{L}\mathbf{x}_k(\boldsymbol{\theta})], \tag{3.4}$$

and $\mathbf{\Sigma} \in \mathbb{R}^{N_{mod} \times N_{mod}}$ is the covariance matrix of the prediction error $\mathbf{e}(\boldsymbol{\theta})$ modelled as a Gaussian random vector with zero mean.

The posterior distribution provides an estimate of the uncertainty in the model parameters, $\boldsymbol{\theta}$ given the information content in the acquired data; therefore, its information entropy provides a unique scalar measure of this uncertainty and is defined by

$$h(\mathbf{L}; \mathbf{\Sigma}, \mathbf{D}) = \mathbb{E}_{\boldsymbol{\theta}}[-\ln p(\boldsymbol{\theta} \mid \mathbf{\Sigma}, \mathbf{D})] = -\int \ln p(\boldsymbol{\theta} \mid \mathbf{\Sigma}, \mathbf{D}) p(\boldsymbol{\theta} \mid \mathbf{\Sigma}, \mathbf{D}) \, d\boldsymbol{\theta}. \tag{3.5}$$

Given a sufficiently large amount of data, $(n \to \infty)$ [26] equation (3.5) can be replaced by an asymptotic approximation. The approximate information entropy is defined as

$$h(\mathbf{L}; \mathbf{\Sigma}, \mathbf{D}) \simeq H(\mathbf{L}; \boldsymbol{\theta}, \mathbf{\Sigma}) = \frac{1}{2} N_{\theta} \ln(2\pi) - \frac{1}{2} \ln[\det \mathbf{Q}(\mathbf{L}; \boldsymbol{\theta}, \mathbf{\Sigma})], \tag{3.6}$$

where $\mathbf{Q}(\mathbf{L}; \boldsymbol{\theta}, \mathbf{\Sigma})$ is a $N_{st} \times N_{st}$ semi-positive definite matrix known as the FIM which contains information about the uncertainty in the estimate of parameters $\boldsymbol{\theta}$ given the measured data from all monitored positions specified in $\mathbf{L}$ and is asymptotically approximated by

$$\mathbf{Q}(\mathbf{L}; \boldsymbol{\theta}, \mathbf{\Sigma}) = \sum_{t=1}^{n} (\mathbf{L} \nabla_{\theta} \mathbf{x}_t(\boldsymbol{\theta}))^T (\mathbf{L} \mathbf{\Sigma} \mathbf{L}^T)^{-1} (\mathbf{L} \nabla_{\theta} \mathbf{x}_t(\boldsymbol{\theta})), \tag{3.7}$$

in which $\nabla_{\theta} \mathbf{x}_t(\boldsymbol{\theta})$ is the gradient of the model response with respect to its parameters $\boldsymbol{\theta}$.

In modal analysis, the response vector $\mathbf{x}(\boldsymbol{\theta})$ can be expressed as a linear combination of the mode shape, $\mathbf{x}(\boldsymbol{\theta}) = \mathbf{\Phi}\boldsymbol{\theta}$, where $\boldsymbol{\theta} = \boldsymbol{\zeta} \in \mathbb{R}^m (m \leq N_{mod})$ is the modal coordinate vector and $\mathbf{\Phi} \in \mathbb{R}^{N_{mod} \times m}$ is the mode shape matrix. Therefore, steering sensors to positions where the covariance of the estimated $\boldsymbol{\zeta}$ is minimal maximizes the information content of the measured responses. The gradient in equation (3.7) is then given as $\nabla_{\theta} \mathbf{x}_t(\boldsymbol{\theta}) = \mathbf{\Phi}$, and substituting back yields

$$\mathbf{Q}(\mathbf{L}; \boldsymbol{\theta}, \mathbf{\Sigma}) \equiv \mathbf{Q}(\mathbf{L}, \mathbf{\Sigma}) = (\mathbf{L}\mathbf{\Phi})^T (\mathbf{L}\mathbf{\Sigma}\mathbf{L}^T)^{-1} (\mathbf{L}\mathbf{\Phi}), \tag{3.8}$$

as presented in [20]. Equation (3.8) shows, therefore, that the FIM is only dependent on the observation matrix $\mathbf{L}$, the mode shape $\mathbf{\Phi}$ and the covariance matrix $\mathbf{\Sigma}$. Note that $\mathbf{L}$ is a non-square matrix, which prevents the trivial simplification of equation (3.8).

(ii) Spatial correlation of prediction error

The spatial correlation of the prediction error is essential for the formulation of the sensor configuration reward at each learning step, as it enforces a minimum distance between sensors within the three-dimensional space. As previously established, the prediction error $\mathbf{e}(\boldsymbol{\theta})$ is evaluated as the sum of the model and measurement error, and if independence is assumed, $\mathbf{\Sigma}$ takes the form

$$\mathbf{\Sigma} = \mathbf{\Sigma}_{mea} + \mathbf{\Sigma}_{mod}, \tag{3.9}$$

where $\mathbf{\Sigma}_{mea}$ represents the measurement error covariance and $\mathbf{\Sigma}_{mod}$ accounts for the model error covariance. It is reasonable to assume measurement [20] errors are location-independent, resulting in a diagonal covariance matrix $\mathbf{\Sigma}_{mea} = c^2 \mathbf{I}$, where c is a variance constant and $\mathbf{I}$ is an identity matrix. In contrast, model errors often exhibit spatial correlation between neighbouring locations owing to underlying model dynamics, requiring a non-diagonal covariance matrix to capture these dependencies. The model error covariance can then be expressed as an exponential function given as

$$\Sigma_{mod,ij} = \sqrt{\Sigma_{ii}\Sigma_{jj}} \exp\left(-\frac{\delta_{ij}}{\delta}\right), \tag{3.10}$$

where the spatial distance between DOFs i and j, denoted as δ_{ij}, is used to characterize the spatial correlation of signals, the parameter δ, known as the correlation length, quantifies the extent of spatial correlation and the auto-correlation terms Σ_{ii} and Σ_{jj} are assumed to be equal to one, representing perfect correlation at similar locations. The cross-correlation term Σ_{ij} spans from zero to one, where $\Sigma_{ij} = 1$, indicates perfect spatial correlation between sensors, and when $\Sigma_{ij} \to 0$, the
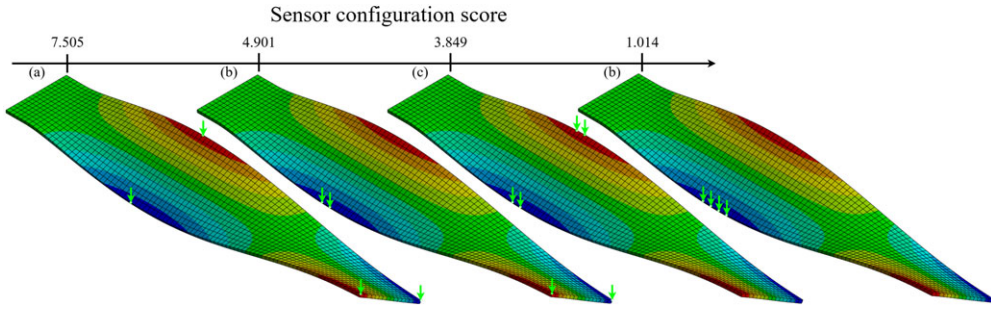
Figure 4. Sensor configurations score for the second torsional mode shape of a clamped cantilever structure showing (a) four sensors with an efficient spatial distribution given a high score, (b) reduced sensor configuration score owing to two close sensors, (c) further reduction in sensor configuration score resulting from two pairs of close sensors and (d) poor spatial distribution of sensors with low configuration score. The reward function takes into consideration the distance between sensors and penalizes configurations with low spatial distribution.

correlation decreases as the distance δ_{ij} between sensors increases. This relationship reflects the diminishing spatial correlation with increasing sensor separation.

However, equation (3.10) has been reported to result in suboptimal reward owing to boundary case placement preference [27]. To address this, we rely on a correlation function [28], such that for the same DOFs i, j, the model error covariance is expressed as

$$\Sigma_{ij} = \exp\left(-\frac{\delta_{ij}}{\delta}\right) \frac{\boldsymbol{\psi}_i^\top \boldsymbol{\psi}_j}{N_M}, \tag{3.11}$$

where $\boldsymbol{\psi}_i = \psi_{k,i} \mid k = 1, \dots, K$ and $\boldsymbol{\psi}_j = \psi_{k,j} \mid k = 1, \dots, K$ are evaluated as

$$\psi_{k,i} = \frac{|\phi_{k,i}|}{\max(|\phi_{k,i}|, |\phi_{k,j}|)} \quad \text{and} \quad \psi_{k,j} = \frac{|\phi_{k,j}|}{\max(|\phi_{k,i}|, |\phi_{k,j}|)}, \tag{3.12}$$

and δ is defined as the ratio of the greatest distance across all DOFs to the total number of sensors. The terms $\phi_{k,i}$ and $\phi_{k,j}$ are the mode shapes at positions i and j for mode k, respectively.

The terms $\psi_{k,i}, \psi_{k,j} : \mathbb{R} \to [0, 1]$ are set to unity when the mode shape component for i and j is zero, indicating full correlation. The product $\boldsymbol{\psi}_i^\top \boldsymbol{\psi}_j : \mathbb{R} \to [0, K]$ provides a mode-shape-dependent weighting on equation (3.10) and Σ_{ii} is assumed to be unity (self-correlation). The robustness of this spatial correlation function is shown graphically in figure 4.

Equation (3.6) shows that minimizing the information entropy is equivalent to minimizing the $det(\mathbf{Q}(\mathbf{L}, \boldsymbol{\Sigma}))$; here, we exploit the semi-positive-definite structure of $\mathbf{Q}(\mathbf{L}, \boldsymbol{\Sigma})$ and instead compute the product of the squares of singular values of the Cholesky factorized $\mathbf{Q}(\mathbf{L}, \boldsymbol{\Sigma})$ equation [28]. The immediate reward R is evaluated as the difference between consecutive sensor configurations, which we refer to as the score in subsequent sections of this paper:

$$R := det(\mathbf{Q}(\mathbf{L}', \boldsymbol{\Sigma}))_{\text{current}} - det(\mathbf{Q}(\mathbf{L}, \boldsymbol{\Sigma}))_{\text{previous}}$$

$$= \left[\prod_{i=1}^{N_{\text{mod}}} \sigma_i^2 \right]_{\text{current}} - \left[\prod_{i=1}^{N_{\text{mod}}} \sigma_i^2 \right]_{\text{previous}},$$

where σ_i is the singular value. The agent receives a positive reward if a change in the position of a sensor within the current configuration results in an increase in the determinant of FIM compared to the previous configuration. The spatial correlation component of the reward function penalizes configurations in which sensors are positioned in close proximity. In cases where two or more sensors occupy the same node, a reward of negative one is assigned, which is substantial relative to the maximum achievable reward within an episode. This mechanism encourages spatially distributed sensor placements and effectively prevents sensor collisions during operation.

(iii) Model dynamics and discounting factor

The model dynamics, also known as the transition probability, defines the likelihood of transitioning from one state to another given a specific action. In the context of the sensor steering problem, the model dynamics is deterministic. This means that given an action a and a state s, the probability of transitioning to a new state s' is 1. For example, if a sensor is moved in a specific direction, the new configuration of the sensors is determined with certainty. At the boundaries of the placement region, the probability of taking an action that moves a sensor outside the region is 0, and the state s remains unchanged.

The discounting factor γ determines the importance of future rewards relative to immediate rewards. A discount factor close to 1 implies that future rewards are nearly as important as immediate rewards, encouraging the agent to consider long-term benefits. Conversely, a discount factor close to 0 places more emphasis on immediate rewards, potentially leading to shortsighted decisions. In the context of the sensor steering problem, the discounting factor helps balance the trade-off between immediate improvements in a single sensor placement and the long-term goal of optimizing the overall sensor configuration. By appropriately tuning the discount factor, whether through optimization or heuristics, the agent can be directed to make decisions that achieve the optimal final configuration.

(c) Reinforcement learning

Reinforcement learning is based on an MDP formulation and involves having an agent interact with an environment to learn a policy which maximizes a specified reward [29]. The sensor steering problem can be viewed as a sequential decision process that involves decisions made by an agent $\pi_t(a,s) : Pr\{A_t = a | S_t = s\}$ about where to move the sensors given the current state of the digital twin. For a robust digital twin framework, it is valuable to make these decisions with a consideration of uncertainty or some measure of risk rather than just expectations. A natural algorithm choice that presents these features is a value-based distributional RL [30]. Here the aleatoric (intrinsic) uncertainty owing to the stochasticity of the policy and environment is modelled by the return distribution. More precisely, we utilize the deep Q-network algorithm called Rainbow [31] which combines significant algorithmic improvements of the original deep Q-network. Rainbow is formulated based on an equivalent distributional Bellman equation given as

$$Z(s,a) \overset{D}{=} R(s,a) + \gamma Z(S',A'), \tag{3.13}$$

where $Z(s,a)$ is the return distribution whose expectation is the value function $Q(s,a)$ which is not related to equation (3.8) and is given as

$$Q(s,a) := \mathbb{E}Z(s,a) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t R(s,a)\right], \tag{3.14}$$

where R is a random variable representing the reward. The learning involves minimizing the Kullback–Leibler divergence between the online distribution Z_t and the target distribution $Z_t^{(q)}$, given as

$$D_{KL}(\Psi_z Z_t^{(q)} || Z_t), \tag{3.15}$$

where Ψ_z is the projection onto Z since the online and target distributions have disjointed support. The return distribution is parameterized by a categorical distribution over a fixed number of points called atoms, which are equidistant and bound by $[V_{\min}, V_{\max}]$ representing the canonical returns.

4. Proposed framework

This section describes how the proposed strategy fits into the digital twin paradigm. The learning environment is considered an abstraction of the virtual environment defined by the
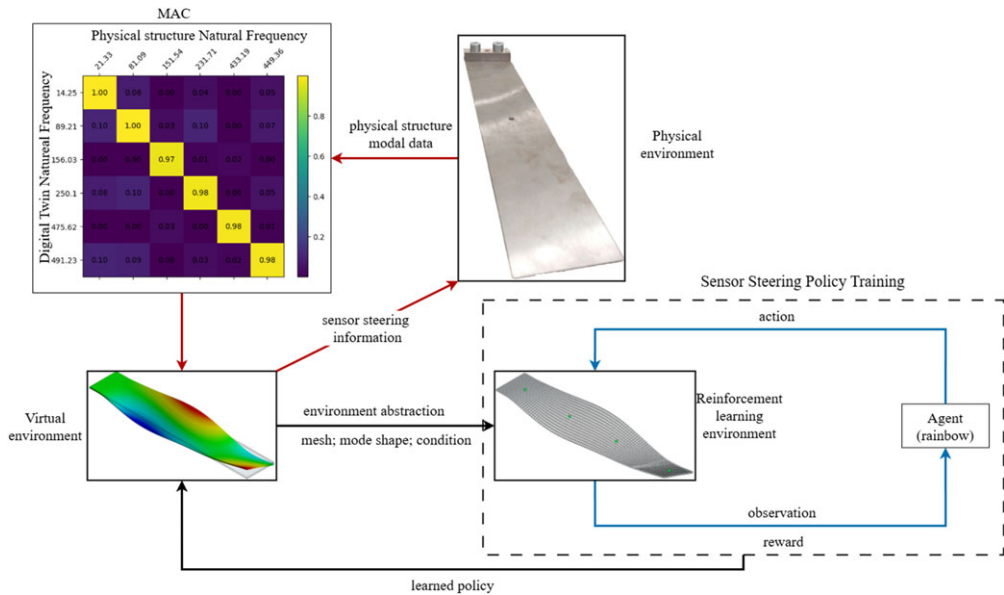
Figure 5. Graphical representation of interaction between physical and digital space. The red cycle indicates the virtual-to-physical and physical-to-virtual interaction; the black cycle shows the interaction between the virtual and RL environments and the blue cycle is the agent–environment interaction.

current structural condition, mesh information, sensor details and mode shape as shown in figure 5. With these data, a custom RL environment is set up using the OpenAI Gymnasium environment framework [25]. An optimal policy is learnt by steering all available sensors within the environment to explore various sensor placement configurations and thereby find the shortest path to the optimal positions. At each environment initialization (reset), all available sensors are evenly distributed along the longitudinal line of symmetry of the structure, representing an heuristic configuration upon which the agent improves (see figure 5). The trained agent is deployed to the virtual space, where it adapts the position of the sensors given a known virtual environment state, and finally, this information is made available to the physical space.

In the physical space, a typical sensor steering step involves sensors that are capable of moving along tracks, such as rails. Given information from the digital twin, these sensors are dynamically repositioned along the rails to new positions of interest. One notable possible application of this approach is in the monitoring of large linear structures such as bridges. By installing rails along the length of the bridge, sensors could be systematically steered to various locations as needed. This strategy not only optimizes data collection but also results in a significant reduction in the number of installed sensors. Instead of deploying a large number of fixed sensors across the entire structure, a smaller set of mobile sensors could cover multiple points of interest, thereby reducing costs and simplifying maintenance.

(a) Implementation details

This paper focuses on the sensor steering strategy, assuming that an instance of the digital twin with a good representation of the physical asset is available. Therefore, we do not include a continuous validation and updating process. The physical structure is a steel cantilever plate with the following dimensions: 447 mm in length, 76.2 mm in width and 3 mm in thickness. The plate is clamped at one end at a depth of 24 mm from the end. A finite element model of the physical structure with an element size of 5 mm and similar boundary conditions is constructed, and the computed modal assurance criterion (MAC) is shown in figure 5 (top left). The MAC

demonstrates a high correlation in the mode shape, which is crucial for computing the learning reward. The MAC values also validate that our simulation model accurately represents the physical structure. To build the virtual environment, the simulation model is scripted into a sequence of commands using the Ansys Pythonic API (PyMechanical). Data from the virtual space are extracted and used to define the custom RL environment. Figure 2 shows the RL environment search region defined by the candidate sensor location. Different options were considered for handling edge location actions. Initially, we defined agent actions outside the search region as episode termination actions, which resulted in resetting the environment. However, this set-up proved to be inefficient and led to short learning episodes. Subsequently, we defined actions outside the search region as null actions, meaning the next state is the same as the previous state. This approach proved to be effective for the learning process. Finally, the Rainbow agent is implemented using the Tianshou Python package [32]. The environment configuration scores are normalized to a maximum of 1 and $\delta = 0.429\,81$, which is the longest length across the cantilever plate placement region. The DRL agent hyperparameters are similar to values in Hessel *et al.* [31] with the exception of $\gamma = 0.9$ and target network period $= 3200$, while the learning rate, priority exponent, and multi-step are case specific. For the exploration strategy, we adopt an $\epsilon-$greedy strategy but anneal ϵ to 0.01 in the first 250 000 steps.

5. Results

In this section, we demonstrate the effectiveness of our proposed framework, described in §3, when applied to a cantilever plate structure. We present two case studies: the first focuses on sensor steering in a structural damage severity setting (severity of the number and locations of damage), and the second on damage localization. Considering the complexity of the placement region as indicated by 1462 candidate sensor locations and their combinations given the number of deployed sensors, we focus on the first three vibration modes, a maximum of four deployed sensors and a library of three damage conditions similar to the model library presented in [33] in both case studies to guarantee the tractability of the problem. Damage is simulated by the addition of point masses, which result in minimal changes in the first mode shape and corresponding significant changes in higher mode shapes. These case studies illustrate the practical application of our framework and its potential to enhance the accuracy and reliability of digital twins in dynamic environments.

(a) Adapting based on damage severities

Our first consideration is a damage severity problem. The goal, therefore, is to learn a policy capable of adapting sensor positions with increasing degrees of damage. We consider two different damage severities, in addition to the healthy condition. The healthy condition represents the baseline state of the cantilever plate without any structural issues. The two damage conditions simulate different scenarios where the plate experiences structural damage. These conditions are illustrated in figure 6.

Set-up:

— *Healthy condition*. The cantilever plate is in its original, undamaged state (figure 6a).
— *Damage severity 1*. A point mass of 0.7 kg simulates a moderate level of damage at the edge of the plate (figure 6b).
— *Damage severity 2*. Two point masses of 0.7 kg each, which simulate an increased level of damage on the same edge side of the plate (figure 6c).

For this case study, the environment is initialized with three sensors as described in §4. The agent is trained using the DRL agent with case-specific hyperparameters of learning rate $= 6.25 \times 10^{-5}$, priority exponent $= 0.5$, multi-step $= 3$ and a fully connected multi-layer perceptron (MLP) with two layers consisting of 256 units and 128 units. To demonstrate the condition-specific
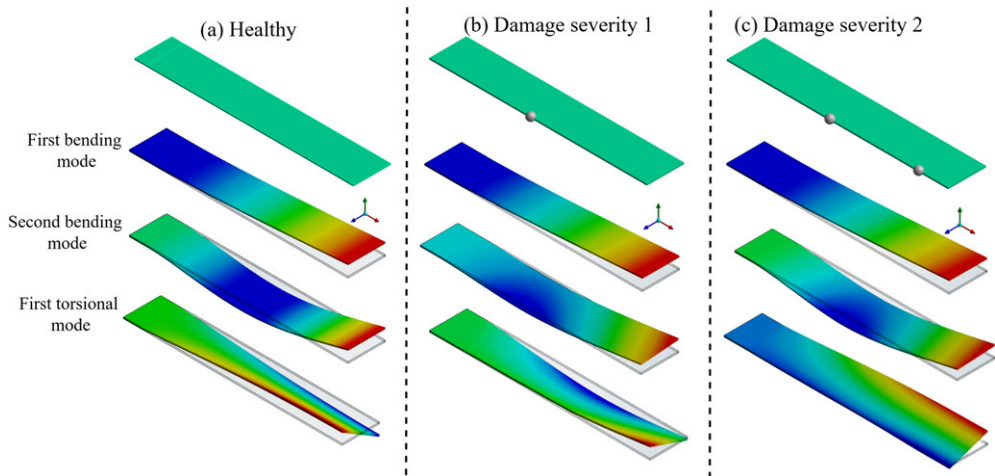
Figure 6. The first three mode shapes of different structural conditions of a cantilever plate showing (a) the healthy condition, (b) damage condition 1 simulated by a point mass of 0.7 kg and (c) damage condition 2 simulated by two point masses of 0.7 Kg on the same edge.
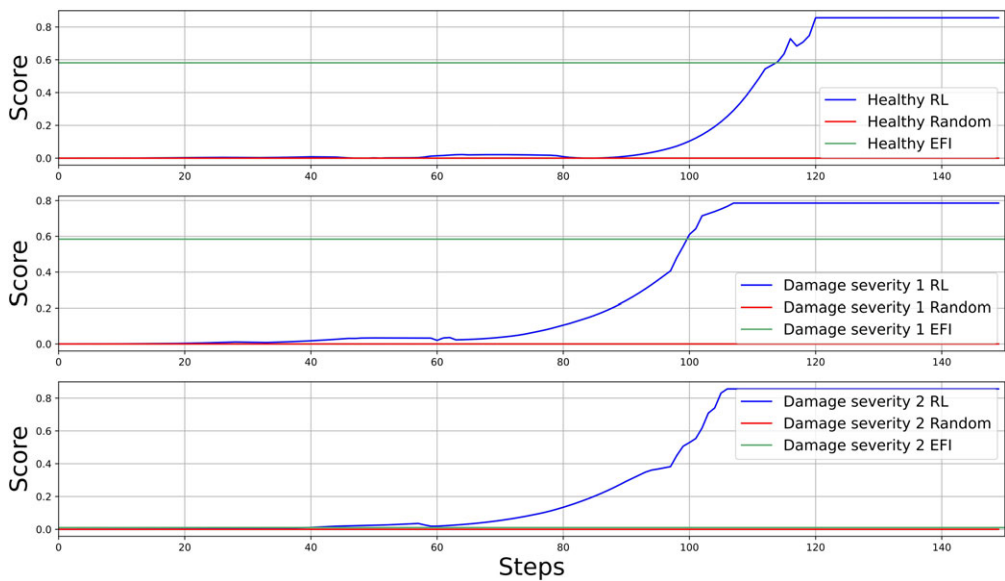

Figure 7. Agent condition-specific evaluation showing scores for healthy (top), damage severity 1 (middle) and damage severity 2 (bottom). The agent takes 0.004 s to execute each step within the environment.

performance, figure 7 presents a comparative analysis of the trained agent against an effective independence methodology [27] with the same error covariance matrix and a random agent as baselines. The results indicate that the trained agent significantly outperforms both baselines within each condition. The trained agent achieves a score above 0.85 in both the healthy and damage severity 2 conditions within 120 and 105 steps, respectively. In the case of damage severity 1, it reaches a score of approximately 0.79, still exceeding the performance of the baselines.

To achieve this performance, we further describe the agent training approach. Figure 8 shows the agent's mean episode reward performance averaged over three test episodes per epoch.
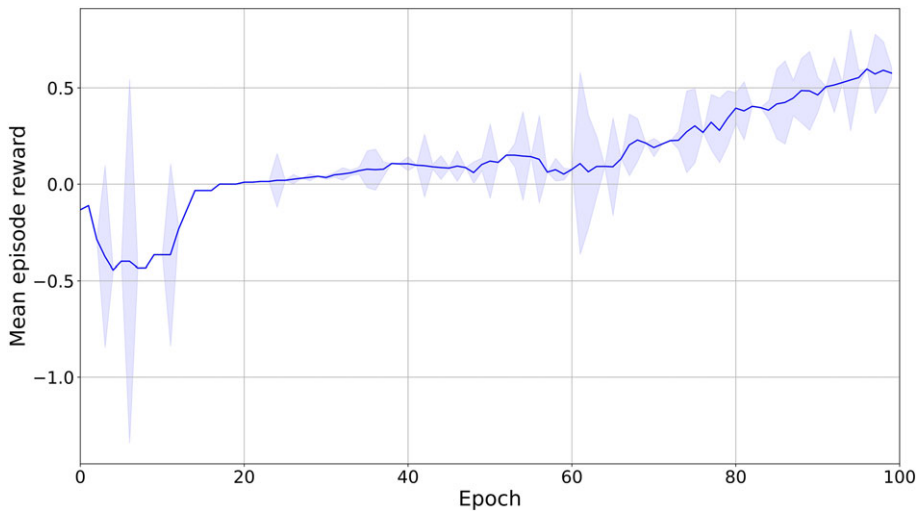
Figure 8. Mean episode reward averaged over three test episodes for healthy and damage severity 1 and 2 with standard deviation (shaded region). Curve is smoothed with a moving average of 10 points.

This indicates the agent's capacity to learn an optimal sensor steering policy across all three structural conditions. The shaded region represents the agent–environment interaction standard deviation. Each training epoch consists of 10 000 steps, and each environment episode terminates after 1000 steps. After each environment termination, a random structural condition is sampled for the next episode to enable the policy to generalize effectively to all structural conditions. This approach is similar to domain randomization technique for sim-to-real problems within the robotic community [34]. The final score performance, which is the final score at the end of each episode averaged over test episodes, is shown in figure 9. The final score started with a low value at the early stages of training and increased to a maximum of 0.8039, demonstrating its effectiveness in accurately repositioning sensors.

Finally, we also show that the agent learns the optimum path to the position that maximizes the configuration score, as described in §3b. Within a test episode, the agent first steers the sensors to their optimal positions and then maintains this final configuration for the remaining steps. Consequently, a typical episode score trajectory is dominated by the final score, making the plot of the sum of a test trajectory score approximately similar to the final score but scaled by the length of the test episode. This proximity is desirable and indicates that in each episode, only a few initial steps are taken by the agent to reach the optimal positions. This similarity property is shown in figure 9.

(b) Adapting to different damage locations

In this case study, we demonstrate the efficacy of the proposed framework in adapting sensors to damage at different locations of a structure. Similar to the previous case, we consider two different damage conditions, in addition to the healthy condition. The goal, therefore, is to learn a policy capable of adapting sensor positions given that damage at specific locations have altered the structure's dynamic behaviour (see figure 10).

To provide a comprehensive analysis, we established the following conditions for the cantilever plate structure:

— *Healthy condition* (figure 10a).
— *Damage conditions 1 and 2*. A point mass of 0.2 kg simulates damage at the two vertices locations of the plate, respectively (figure 10b,c).
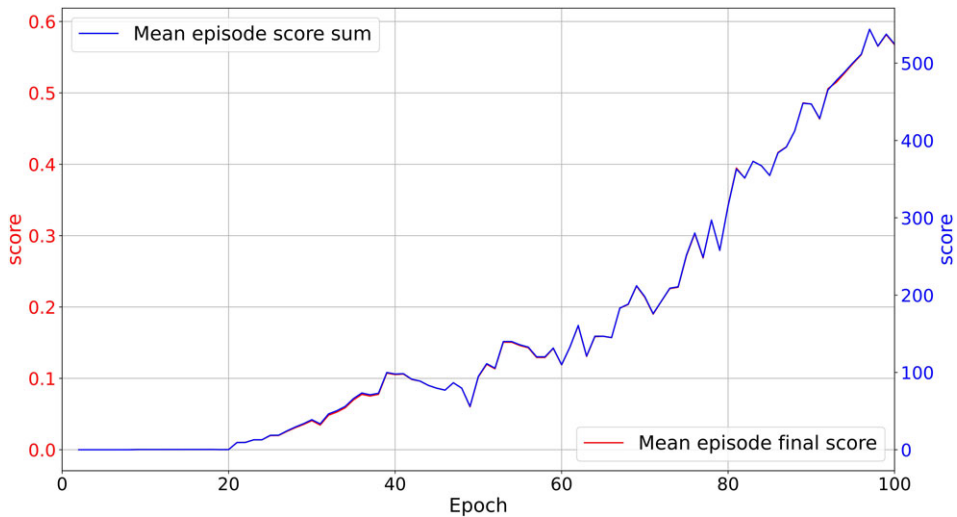
Figure 9. Score metric averaged over three test episodes per epoch for healthy and damage severity 1 and 2: episode score sum (red) and episode final score (blue). Similarity of plots shows the agent's performance in following the optimal path to the optimal location. Curve is smoothed with a moving average of 10 points.
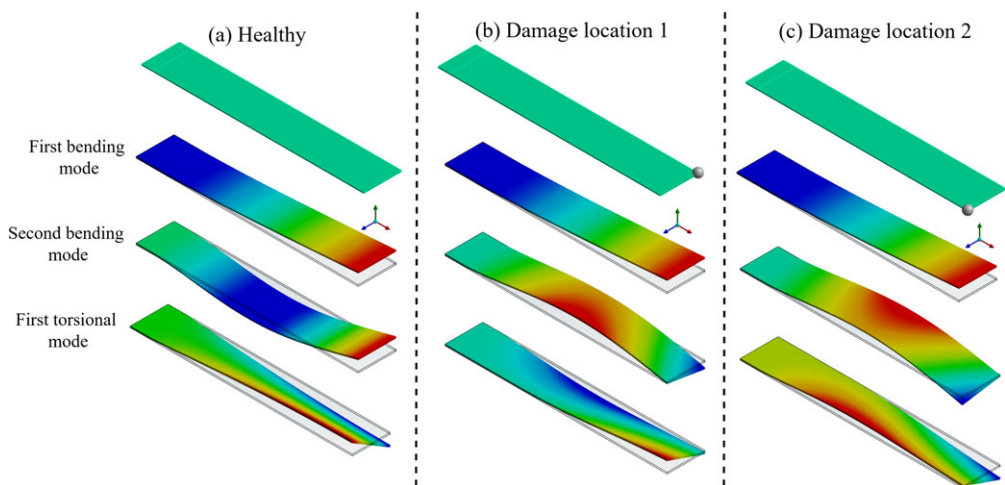

Figure 10. The first three mode shapes of different structural conditions of a cantilever plate showing (a) the healthy condition, (b) damage condition 1 simulated by a point mass of 0.2 kg at the right-hand vertex and (c) damage condition 2 simulated by a point mass of 0.2 kg at the left-hand vertex.

In this case, we use four sensors, and the agent is hyperparameterized with a learning rate of 1.25×10^{-4}, a priority exponent of 0.7, multi-step parameter of 5 and MLP with two layers of 128 units each. The agent is then trained using a similar approach as the previous case but for 200 epochs to ensure comprehensive learning and generalization. We now evaluate the condition-specific performance of the agent. Figure 11 shows the performance compared to baselines described in §5b. The trained agent outperforms the baselines and finds the optimal location after fewer than 150 steps in the healthy condition and damages location 1 condition. In the final condition (damage location 2), the agent achieves suboptimal results of 0.4, although still better than the baselines. The agent's action trajectory in this condition is visualized in figure 12, and
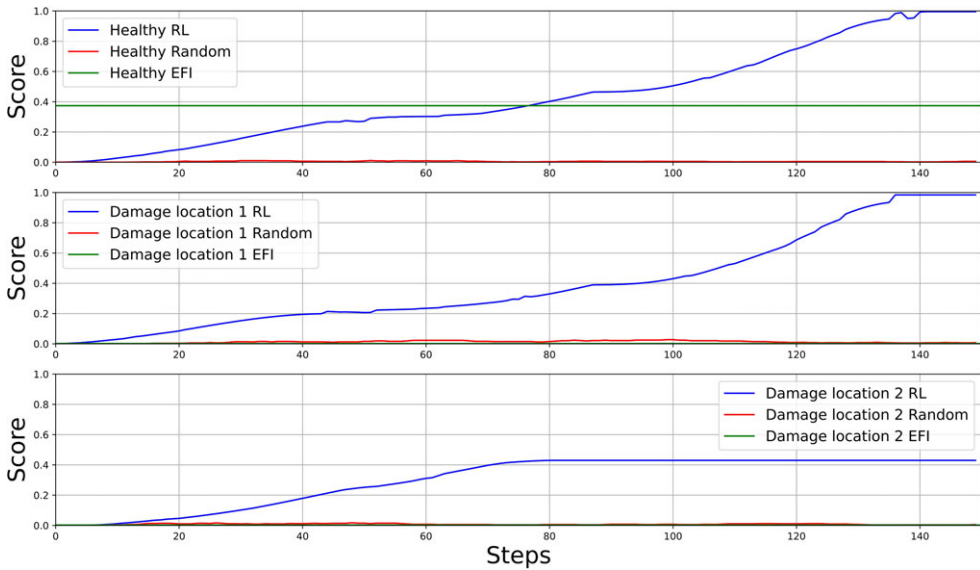
Figure 11. Agent condition-specific evaluation showing scores for healthy (top), damage location 1 (middle) and damage location 2 (bottom). The agent takes 0.004 s to execute each step within the environment.
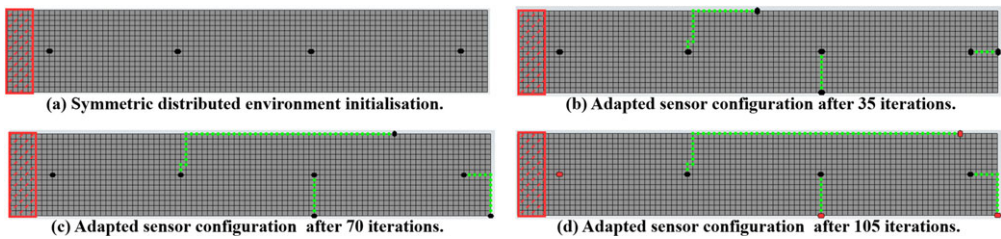

(a) Symmetric distributed environment initialisation.

(b) Adapted sensor configuration after 35 iterations.

(c) Adapted sensor configuration after 70 iterations.

(d) Adapted sensor configuration after 105 iterations.

Figure 12. Trained agent adapts sensor configuration for damage location 2 condition showing agents action path from (a) initialization and after (b) 35 steps, (c) 70 steps and (d) 105 steps. The agent performance in this condition is attributed to the first sensor being fixed throughout the agent training process, as depicted by the final sensor positions (red marker).

it can be seen that the first sensor remains unchanged, indicating that the agent during training learned nothing about it. This can be attributed to the increase in learning complexity introduced by one extra sensor.

During the training process, the agent's performance was also evaluated based on several metrics. The primary performance metric (mean episode reward), illustrated in figure 13, shows that the agent achieved a maximum mean reward of 0.9982 which was reached at the final training epoch, indicating consistent learning outcomes. In addition to the reward performance, we also assessed the agent's performance score and the trajectory towards optimal sensor positioning. As shown in figure 14, the agent achieved a maximum score of 0.9839, demonstrating its effectiveness in accurately repositioning sensors. Figure 14 also shows a degradation in the performance of the agent between epochs 100 and 125; from our observation, this is due to increased sampling of the damage severity 2 condition during environmental initialization, which results in suboptimal performance. The path to optimal sensor position performance is detailed in figure 14, which, as explained previously, is similar to the final score plot and whose similarity indicates optimal path identification performance.
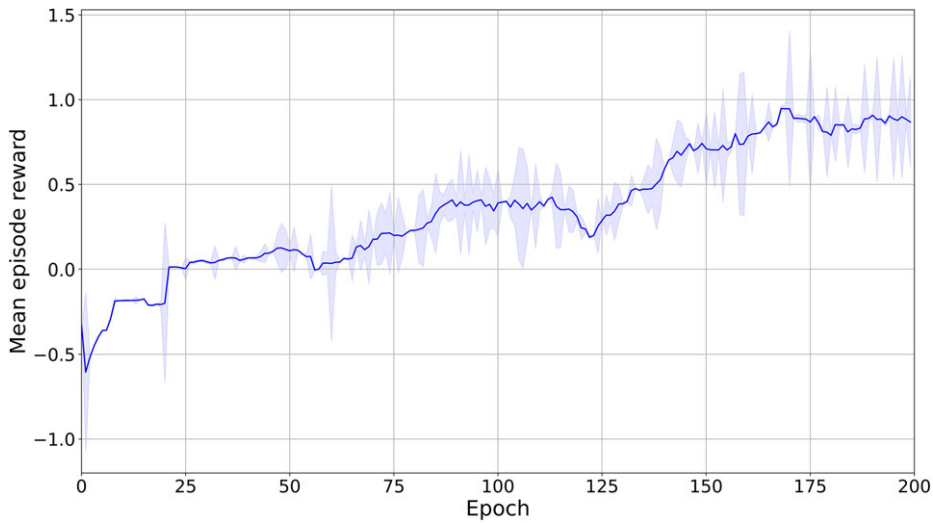
Figure 13. Mean episode reward averaged over three test episodes for healthy and damage locations 1 and 2 with standard deviation (shaded region). Curve is smoothed with a moving average of 10 points.

Figure 14. Score metric averaged over three test episodes per epoch for healthy and damage locations 1 and 2: episode score sum (red) and episode final score (blue). Similarity of plot shows the agent's performance in following the optimal path to the optimal location. Curve is smoothed with a moving average of 10 points.

These cases demonstrate the practical application of the proposed framework in adapting the sensor locations to structural damage. The results serve as a benchmark for evaluating strategies aimed at dynamically adjusting sensor positions based on real-time feedback for timely identification and assessment of potential issues.

6. Conclusion

In this study, we have presented a novel approach for dynamic data acquisition in digital twins by employing adaptive sensor steering strategies based on DRL. This work is a first step towards adaptive sensing for improved structural monitoring. The proposed framework is motivated by the need for continuous assimilation of informative data relevant to the digital twin at

various stages of its operational lifespan, a requirement for which existing OSP techniques are not designed. The problem is formulated as an MDP and implemented with a value-based distributional RL agent. We present a new custom reinforcement environment architecture with different damage scenarios extracted from the digital twin, and an agent is trained to act optimally by adapting the sensor positions when an observed damage condition is encountered. The trained agent is deployed in the digital twin to guide data acquisition. The proposed framework was validated through two cases on a cantilever plate structure under various conditions, including healthy and damaged conditions. Our results demonstrate the capability of the DRL-based policy to adaptively reposition sensors, hence improving the accuracy and reliability of the digital twin's representations. Both cases assume that an updated digital twin of the physical asset is available and therefore ignore the continuous updating process, which is essential in practice and should be integrated. The proposed framework can also be used for acquiring relevant data at the design stage of a digital twin (see figure 1).

The proposed approach faces two key limitations: (i) the performance of the agent degrades significantly with the increasing number of modes considered, sensors available, and damage cases integrated. This is the curse of dimensionality. (ii) Similar to [22], we assume the damage states are known and fixed, which is impractical in many applications. However, the proposed methodology presents an opportunity to learn and interpolate between conditions seen during training.

A key observation during this study is the sensitivity of agent performance to the exploration strategy. Consequently, future work will focus on developing an exploration strategy specific to the problem, which allows for expedited learning and improved performance at higher levels of difficulty. Moreover, the decision-making process in the current study is based on the expectation of the value distribution. Exploring the decision-making process using other metrics is an area of future interest. In addition, future research will examine the kinematic effects of sensor mobility on host structures, with the goal of developing optimized steering strategies that maximize sensing capability while minimizing any adverse influence on the structural dynamics.

References

1. Wagg DJ, Burr C, Shepherd J, Conti ZX, Enzer M, Niederer S. 2025 The philosophical foundations of digital twinning. *Data-Centric Eng.* **6**, e12. (doi:10.1017/dce.2025.4)

2. Ferrari A, Willcox K. 2024 Digital twins in mechanical and aerospace engineering. *Nature Comput. Sci.* **4**, 178–183. (doi:10.1038/s43588-024-00613-8)

3. Corral-Acero J *et al.* 2020 The 'Digital Twin' to enable the vision of precision cardiology. *Eur. Heart J.* **41**, 4556–4564. (doi:10.1093/eurheartj/ehaa159)

4. Isaac T, Petra N, Stadler G, Ghattas O. 2015 Scalable and efficient algorithms for the propagation of uncertainty from data through inference to prediction for large-scale

problems, with application to flow of the Antarctic ice sheet. *J. Comput. Phys.* **296**, 348–368. (doi:10.1016/j.jcp.2015.04.047)

5. Wagg D, Worden K, Barthorpe R, Gardner P. 2020 Digital twins: state-of-the-art and future directions for modeling and simulation in engineering dynamics applications. *ASCE-ASME J. Risk Uncert. Eng. Syst., Part B: Mech. Eng.* **6**, 030901. (doi:10.1115/1.4046739)

6. Cicirello A. 2024 Physics-Enhanced Machine Learning: a position paper for dynamical systems investigations. (http://arxiv.org/abs/2405.05987)

7. Willcox K, Segundo B. 2024 The role of computational science in digital twins. *Nature Comput. Sci.* **4**, 147–149. (doi:10.1038/s43588-024-00609-4)

8. Hassani S, Dackermann U. 2023 A systematic review of optimization algorithms for structural health monitoring and optimal sensor placement. *Sensors* **23**, 3293. (doi:10.3390/s23063293)

9. Tan Y, Zhang L. 2020 Computational methodologies for optimal sensor placement in structural health monitoring: A review. *Struct. Health Monit.* **19**, 1287–1308. (doi:10.1177/1475921719877579)

10. Wang Y, Chen Y, Yao Y, Ou J. 2023 Advancements in optimal sensor placement for enhanced structural health monitoring: current insights and future prospects. *Buildings* **13**, 3129. (doi:10.3390/buildings13123129)

11. Udwadia FE. 1994 Methodology for optimum sensor locations for parameter identification in dynamic systems. *J. Eng. Mech.* **120**, 368–390. (doi:10.1061/(ASCE)0733-9399(1994)120:2(368))

12. Salama M, Rose T, Garba J. 1987 Optimal placement of excitations and sensors for verification of large dynamical systems. In *28th Structures, Structural Dynamics and Materials Conference*, p. 782.

13. Meo M, Zumpano G. 2005 On the optimal sensor placement techniques for a bridge structure. *Eng. Struct.* **27**, 1488–1497. (doi:10.1016/j.engstruct.2005.03.015)

14. Chang M, Pakzad SN. 2014 Optimal sensor placement for modal identification of bridge systems considering number of sensing nodes. *J. Bridge Eng.* **19**, 04014019. (doi:10.1061/(ASCE)BE.1943-5592.0000594)

15. Li B, Li D, Zhao X, Ou J. 2012 Optimal sensor placement in health monitoring of suspension bridge. *Sci. China Technol. Sci.* **55**, 2039–2047. (doi:10.1007/s11431-012-4815-8)

16. Yang C. 2021 An adaptive sensor placement algorithm for structural health monitoring based on multi-objective iterative optimization using weight factor updating. *Mech. Syst. Signal Process.* **151**, 107363. (doi:10.1016/j.ymssp.2020.107363)

17. Wang J, Law S, Yang Q. 2014 Sensor placement method for dynamic response reconstruction. *J. Sound Vib.* **333**, 2469–2482. (doi:10.1016/j.jsv.2013.12.014)

18. Papadimitriou C, Beck JL, Au SK. 2000 Entropy-based optimal sensor location for structural model updating. *J. Vib. Control* **6**, 781–800. (doi:10.1177/107754630000600508)

19. Bhattacharyya P, Beck J. 2020 Exploiting convexification for Bayesian optimal sensor placement by maximization of mutual information. *Struct. Control Health Monitor.* **27**, e2605. (doi:10.1002/stc.2605)

20. Papadimitriou C, Lombaert G. 2012 The effect of prediction error correlation on optimal sensor placement in structural dynamics. *Mech. Syst. Signal Process.* **28**, 105–127. (doi:10.1016/j.ymssp.2011.05.019)

21. Jabini A, Johnson EA. 2022 A deep reinforcement learning approach to sensor placement under uncertainty. *IFAC-PapersOnLine* **55**, 178–183. (doi:10.1016/j.ifacol.2022.10.508)

22. Kapteyn MG, Willcox KE. 2022 Design of digital twin sensing strategies via predictive modeling and interpretable machine learning. *J. Mech. Des.* **144**, 091710. (doi:10.1115/1.4054907)

23. Kapteyn MG, Pretorius JV, Willcox KE. 2021 A probabilistic graphical model foundation for enabling predictive digital twins at scale. *Nature Comput. Sci.* **1**, 337–347. (doi:10.1038/s43588-021-00069-0)

24. Tezzele M, Carr S, Topcu U, Willcox KE. 2024 Adaptive planning for risk-aware predictive digital twins. (http://arxiv.org/abs/2407.20490)

25. Towers M *et al.* 2024 Gymnasium: a standard interface for reinforcement learning environments. (http://arxiv.org/abs/2407.17032)

26. Papadimitriou C. 2004 Optimal sensor placement methodology for parametric identification of structural systems. *J. Sound Vib.* **278**, 923–947. (doi:10.1016/j.jsv.2003.10.063)

27. Kim SH, Cho C. 2024 Effective independence in optimal sensor placement associated with general Fisher information involving full error covariance matrix. *Mech. Syst. Signal Process.* **212**, 111263. (doi:10.1016/j.ymssp.2024.111263)

28. Vincenzi L, Simonini L. 2017 Influence of model errors in optimal sensor placement. *J. Sound Vib.* **389**, 119–133. (doi:10.1016/j.jsv.2016.10.033)
29. Barto AG. 2021 Reinforcement learning: An introduction. by Richard's sutton. *SIAM Rev.* **6**, 423.
30. Bellemare MG, Dabney W, Munos R. 2017 A distributional perspective on reinforcement learning. In *International conference on machine learning*, pp. 449–458. PMLR.
31. Hessel M, Modayil J, Van Hasselt H, Schaul T, Ostrovski G, Dabney W, Horgan D, Piot B, Azar M, Silver D. 2018 Rainbow: Combining improvements in deep reinforcement learning. In *Proc. of the AAAI Conf. on Artificial Intelligence*, vol. 32.
32. Weng J, Chen H, Yan D, You K, Duburcq A, Zhang M, Su Y, Su H, Zhu J. 2022 Tianshou: a highly modularized deep reinforcement learning library. *J. Mach. Learn. Res.* **23**, 1–6.
33. Kapteyn MG, Knezevic DJ, Huynh D, Tran M, Willcox KE. 2022 Data-driven physics-based digital twins via a library of component-based reduced-order models. *Int. J. Numer. Methods Eng.* **123**, 2986–3003. (doi:10.1002/nme.6423)
34. Tobin J, Fong R, Ray A, Schneider J, Zaremba W, Abbeel P. 2017 Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ Int. Conf. Intelligent Robots Syst. (IROS)*, pp. 23–30. IEEE.