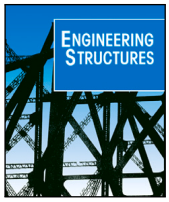
Engineering Structures 274 (2023) 115122

Contents lists available at ScienceDirect

Engineering Structures

journal homepage: www.elsevier.com/locate/engstruct



Active structural control framework using policy-gradient reinforcement learning

Soheila Sadeghi Eshkevari [a,*], Soheil Sadeghi Eshkevari [b], Debarshi Sen [a,c], Shamim N. Pakzad [a]

[a] Department of Civil and Environmental Engineering, Lehigh University, Bethlehem, PA, USA
[b] Senseable City Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA
[c] School of Civil, Environmental, and Infrastructure Engineering, Southern Illinois University, Carbondale, IL, USA

ARTICLE INFO

Keywords:
Active control
Reinforcement learning
Policy-gradient methods
Nonlinear dynamics

ABSTRACT

This paper presents a novel data-driven approach for active structural control through the use of deep reinforcement learning, wherein, the control system *learns* to react in an optimal manner through a training process that utilizes deep neural networks within a reinforcement learning framework. The key advantage of this paradigm is the data-driven approach to active control which helps circumvent the need for high-fidelity modeling that typically requires extensive prior knowledge about the structure of interest. Furthermore, the proposed framework is applicable for designing a variety of active controllers, and different external load types, for example, wind and seismic loads for any desired building. The efficacy of the proposed framework is demonstrated in the context of seismic response control through three numerical case studies. The results confirm that the proposed approach yields significant structural response reductions in the linear and nonlinear regimes. Furthermore, implementation issues such as sensitivity to structural property variations, and time delay are thoroughly investigated.

1. Introduction

Structural systems are designed in accordance to design codes such that they withstand operational and extreme loads estimated based on acceptable performance and risk. However, structures designed through the traditional approach often require supplemental devices to meet demands resulting from exogenous loads that may lead to loss of functionality, collapse and in many cases loss of lives. To enhance the existing process, a considerable effort has been made to transform design philosophy to include life-cycle costs, environmental sustainability, and resiliency of the built environment. In the context of enhancing resiliency, structural control has been a long-standing solution for vibration mitigation in structural systems [1]. In this regard, structures are typically equipped with passive mechanical devices that modify strength, stiffness, and damping which are important factors to counteract demands. However, passive control systems generally undergo residual deformations and require re-tuning as structural properties change over time. Active control systems have been popular in mechanical and aerospace engineering for vibration control and are widely used in vehicles and aircrafts [2–5]. In structural engineering, active control has been proposed and practically experimented; yet, due to the sensitivity and dimensional complexity of the problem, it requires extensive studies prior to ubiquitous field implementation [1]. In addition, there have been concerns regarding the capability of the active control methods in performing robustly in noisy and uncertain environments [6,7].

Despite the challenges, active control systems have the potential to overcome the limitations of its passive counterparts. In the active control scenario, the controllers (e.g., actuators in a building) react in accordance to a computer program that processes real-time structural response measured by sensors. In other words, in active control systems, the agents are integrated in a feedback loop and apply reactions based on the instantaneous demands. However, the majority of the existing active control approaches require analytical solutions based on optimal control theory that requires a full description of the system [8,9]. For instance, optimal control algorithms such as linear quadratic Gaussian (LQG) control require the characteristic matrices of the state–space model of a system. This is a limiting constraint for existing structures as well as for evolving systems (e.g., due to deterioration) since the mechanical properties are not fully known. Consequently, such solutions are prone to instability when the real-world structure deviates from the modeling assumptions. Therefore, an adaptive data-driven solution is highly desired.

In recent years, through advancements in high-performance computing and sensing technology, data-driven methods have received

* Corresponding author.
E-mail address: sos318@lehigh.edu (S. Sadeghi Eshkevari).

https://doi.org/10.1016/j.engstruct.2022.115122
Received 6 March 2022; Received in revised form 5 September 2022; Accepted 9 October 2022
Available online 27 October 2022
0141-0296/© 2022 Elsevier Ltd. All rights reserved.

significant attention and have been successfully implemented for solving a plethora of problems in a wide range of areas, from engineering to natural sciences. Methodologies such as deep learning (DL) and reinforcement learning (RL) have enabled solutions to problems that were considered computationally intractable in image processing, robotics, natural sciences, language processing, to name a few. Of particular interest is the use of RL in adaptive control in robotic systems [10,11]. In this framework, the control problem is formulated as a sequential decision-making modeled with a Markov decision process (MDP). For low-complexity and discrete systems, an optimal policy can be found (e.g. in the context of active control, the policy determines the control forces) by constructing a state–action value table for the MDP, and taking the most valued action in each state of the system. This approach termed as dynamic programming, however, is not widely applicable for real-world tasks due to the unknown nature of the state transition probabilities and closed-form reward functions. Furthermore, in complex and continuous state–action spaces, the closed-form approach would be intractable. However, with the advancements in DL and high performance computing, deep RL (RL functions modeled as deep neural networks) has been widely utilized as a data-driven approach for estimating the optimal policy in MDPs addressing many of the above-mentioned challenges.

Active control has the potential to become a key technology for structural response control if facilitated with data-driven paradigms such as RL. The limitations of existing structural control approaches such as sensitivity to uncertainties, dependence on structural system properties and other practical implementation issues such as power loss and data transmission delays can be addressed by data-driven approaches. Although soft computing tools such as neural networks have been employed to address some of these issues in the past, their application was limited by the computational effort necessary for real world high dimensional problems. Furthermore, existing methodologies focus mostly on the system modeling rather than the decision-making process of an active control system. This strongly motivates the application of RL for this class of problems. However, the existing literature is limited to special cases that do not address many implementation issues such as limited measurements of structural response, scalability and control system's time delay [12]. This paper presents a scalable form of RL-based active control system termed as RL-Controller. Numerical case studies demonstrate the efficacy of the proposed framework when compared to traditional optimal control strategies such as LQG. The key contributions of this work are as follows:

- A flexible and standardized environment for designing active control systems is proposed. The environment models the control problem as an MDP such that the estimation of the control forces is performed by the RL agent. The proposed framework enables the user to define a variety of structures, control mechanisms, layouts, and external loading scenarios. In addition, a comprehensive list of the popular RL optimization algorithms can be simply integrated with this environment which expands its applicability.
- An adaptive composite reward function is defined for training that stabilizes the relative weights of the reward terms (including inter-story drift, acceleration, and applied control force) leading to an improved control performance.
- The proposed active control framework is evaluated on linear and nonlinear structures once trained with artificially designed external loads which can precisely mimic characteristics of actual loads that a structure may be subjected to instead of a Gaussian white noise.
- To address implementation concerns, the RL-based control system is evaluated for different structural property variations and time delay scenarios. It is shown that the proposed algorithm behaves robustly to stiffness, mass, and damping variations and also random and consistent delays with no need for explicit input-state estimations or holding simplifying assumptions.

In the following sections a brief review of active structural control and reinforcement learning is provided. Subsequently, RL-Controller's implementation and results from the numerical case studies involving linear and nonlinear cases are discussed.

2. Active structural control

Deploying structural control devices is an effective means for mitigating the impact of exogenous loads such as seismic and wind loads on buildings and bridges [1,13]. These devices are classified as either passive, semi-active or active control systems. Passive systems such as base isolation systems [14–18], visco-elastic dampers [19–22], and tuned mass dampers [23–27] are well studied and have been used extensively in many parts of the world. However, these systems typically are not designed to account for changes in system properties and load characteristics over time. Active and semi-active control systems address the shortcoming of passive devices. Both active [28,29] and semi-active systems [30] (such as magneto-rheological dampers [31–33], variable [34] and negative stiffness systems [35–37], and tuned mass [38] and liquid dampers [39–41]) estimate control forces based on structural responses in real-time. This typically requires external power sources for active control systems, and special mechanisms for semi-active systems, that allow for an adaptive control force generation. In this paper the focus is on active structural control.

Active control devices generate control forces that help regulate structural responses. Typical control devices deployed on structures include active tendon systems [42], active mass dampers [43], and active viscous dampers [44] to name a few. The generated control force is estimated based on measurements of the external loads and/or structural response [45]. Optimal control algorithms such as linear quadratic regulators (LQR) and linear quadratic Gaussian (LQG) controllers have been successfully applied in the context of civil infrastructure [46]. As the name suggests, LQR is used to optimally control a linear dynamic system with a quadratic cost function that is parameterized by the full state–space. LQG is an extension of LQR, wherein, the assumption related to measurement of the full state–space is relaxed. The relaxation is however limited by the satisfaction of the observability criterion. LQG uses a Kalman filter for estimating the full state–space from a set of measurements and attempts to satisfy the optimal control objective in parallel. The major advantage of these algorithms is the existence of closed-form solutions for the design of the controller.

Other algorithms that have been implemented for active structural control over the years include H_∞, pole placement and sliding mode control (SMC) [47]. The majority of these approaches minimize a cost function parameterized by a set of structural responses and estimated control forces acting on a structure. These algorithms were further enhanced by the inclusion of signal processing frameworks such as wavelet analysis [48], and soft computing tools like fuzzy logic, genetic algorithms and neural networks to deal with cases involving time-varying behavior, nonlinearities and uncertainties [49,50]. While these enhancements do not guarantee optimality, they help develop more versatile and robust control systems. In addition to neural network-based approaches, recently there have been applications of statistical learning techniques such as regression trees [51] and random forests [52]. However, such techniques are limited by the accuracy of system identification performed for developing surrogate models of the system of interest.

Although soft computing tools such as neural networks functioning as model-free alternatives to traditional control have demonstrated their efficacy, they were limited to low dimensional problems owing to the curse of dimensionality and limited computational resources. As discussed earlier, recent advancements in high performance computing have revolutionized the applications of deep neural networks to solve complex problems that were deemed computationally intractable earlier. In this context, RL with embedded deep neural networks emerges as an efficient means to achieve vibration control in engineering systems.

3. Reinforcement learning

RL is a behavioral psychology inspired class of algorithms to solve sequential decision-making problems in various fields of science and engineering wherein the goal is to meet required performance criteria by frequent interaction with the system under uncertainties. The key idea is for an agent to gather experience about a given environment and make informed decisions through actions that help attain the relevant performance criteria, known as agent's exploration and exploitation in an environment. These criteria are defined in terms of a reward function such that favorable actions lead to increased rewards, ensuring that the agent acts to maximize the total reward over a finite or infinite trajectory (in case of infinite or sizable episodes, the total discounted reward is considered with discount factor γ, i.e., the weights of actions constituting the reward function changes over time with higher impact on immediate rewards compared to temporally distant ones). In an RL paradigm, the sequential decision-making problem for the associated environment is formulated as an MDP. The outcome is an optimal policy defined by a probability distribution of actions conditioned on the state of an environment [53]. Deep RL is an extension of RL such that the agent uses a deep neural network that approximates the expected value of actions in different states. This mechanism of learning is more suited for high-dimensional and continuous action space problems. For a comprehensive overview of deep RL, the readers can refer to François-Lavet et al. [54] and Li [55].

Deep RL has been employed in a broad spectrum of fields. Some of the most popular applications are in the areas of autonomous vehicles [56], and for achieving superhuman performance in playing popular games such as Chess and Go [57]. In robotics, deep RL is extensively used for motion control and learning domain specific robotic motions such as balancing and surgical moves [58]. Deep RL has been widely used for multi-agent systems wherein autonomous agents learn to communicate and cooperate in order to solve complex tasks [59]. Deep RL has also been used to facilitate the development of next generation communication paradigms such as 5G, as well as, for solving routing and resource sharing problems in networks [60]. Furthermore, deep RL has applications in genomics, medical imaging and human–robot interfaces [61]. In transportation engineering, deep RL has been studied as an effective, demand adaptive approach for vehicle routing problem [62]. Finally, deep RL has been broadly employed in control problems as well.

RL has significant potential for applications in control problems [63,64]. It has been applied in control engineering in the context of mechanical systems such as vehicles [65], shape control in tensegrity structures [66], as well as in robotics [67]. However, RL in its traditional form was limited to low-dimensional problems. In the past decade, equipped with better computational resources and big data, RL has been applied to far more complicated control problems such as active flow control in computational fluid dynamics [68], bluff body flow control [69], robotic locomotion [70], vision-based robotic manipulation [71], and mapless robot navigation [72], to name a few.

In the context of active structural control, there have been limited studies on the application of RL. For example, RL was recently used to tune a fuzzy logic control-based active mass damper [73]. However, this study was limited to specifically a fuzzy controller-based AMD and did not harness the full potential of RL. Also, the reward function included velocity as a penalty term that can be difficult to measure in a practical sense. Another recent study [74] demonstrated the efficacy of RL for seismic response control of a simple single bay, single story moment frame structure. The RL state that was used in this study included the full state–space of the system, which is again difficult to measure in a real life scenario.

In this work, RL-Controller is proposed; a framework that generalizes the application of RL for structural control problems. In particular, this study is on multi-degree of freedom systems considering the possibility of multiple actuators wherein states and reward functions are defined incorporating limitations in the available measurements in a practical scenario.

4. Methodology

4.1. Reinforcement learning setup

An RL framework in general consists of an environment and one or more agents (Fig. 1(a)). The goal of the agent is to navigate through the various states of a given environment by undertaking a sequence of actions such that a set of performance criteria is satisfied optimally. During training, the agent's policy is repeatedly evaluated by a reward function based on the fitness of the actions it takes. Hence, an RL paradigm strives to obtain an optimal policy defined as a set of actions conditioned on the state of the environment that maximizes the trajectory of rewards. This sequence implies an RL framework consists of four fundamental components: (a) agent, (b) state, (c) reward, and (d) action. Fig. 1(a) shows a general RL framework.

Any RL paradigm learns a policy function that enables the RL agent to take optimal actions given a state of the environment. This policy is typically in the form of a conditional probability distribution $\pi(a|s)$, where a and s are the action and the state of the environment, respectively. This paper focuses on the class of model-free RL algorithms. In this class, the optimal policy is learned without explicitly learning the underlying principles and dynamics of the environment. This is advantageous in this work as it bypasses the complications of building a surrogate model of the dynamic system and instead directly aims for the optimal strategy. To estimate an optimal policy, a popular approach is the construction of a quality function (the expectation of the quality function is also referred to as a state–action value function), $Q(s,a)$. In essence, this function represents the utility of a given action towards expected long-term reward maximization for a given environmental state; estimation of this function is referred to as Q-learning in the literature [75].

Q-learning can be implemented in both on-policy and off-policy settings. The on-policy variant employs the existing best policy available for generating a set of actions for further learning while in the off-policy learning, the updated policy cannot be exploited. The standard Q-learning is limited to discrete action and state spaces. To upgrade the algorithm for continuous state–action spaces, neural function approximators have been proposed. An alternative learning approach is using policy gradient method that directly estimates the optimal policy probability distribution. Policy gradient methods have experienced an increasing popularity in recent years due to the fact that unlike the Q-learning methods, these methods directly learn the action policy and do not rely on the intermediate step of value function learning. In practice, policy gradient methods improve their policy and the value function simultaneously during the learning process. In summary, the optimal policy, the policy objective function and the policy gradient are introduced as shown in Eqs. (1), (2) and (3):

$$\theta^* = \arg\max E_{\tau\sim\pi_\theta(\tau)}\left[\sum_{t=1}^{T} r(s_t,a_t)\right] \tag{1}$$

$$\mathcal{J}(\theta) = E_{\tau\sim\pi_\theta(\tau)}\left[\sum_{t=1}^{T} r(s_t,a_t)\right] \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T} r(s_{i,t},a_{i,t}) \tag{2}$$

$$\nabla_\theta \mathcal{J}(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a_{i,t}|s_{i,t})\right)\left(\sum_{t=1}^{T} r(s_{i,t},a_{i,t})\right) \tag{3}$$

where T is total time of an episode. s, r, and a respectively represent state, reward, and action. θ^* is the optimal set of parameters that construct the optimal policy $\pi_\theta(.)$ such that it maximizes the expected sum of rewards given any possible state. τ represents an episode trajectory within training process. Aligned with this notion, the parameterized objection function $\mathcal{J}(\theta)$ is to be maximized where N is the number of episodes that have been explored. The gradient of this function is also shown as $\nabla_\theta \mathcal{J}(\theta)$. Given the gradient, the optimization problem is simply using the gradient ascent method to maximize function $\mathcal{J}(\theta)$. Policy gradient methods are limited by gradient estimation issues associated

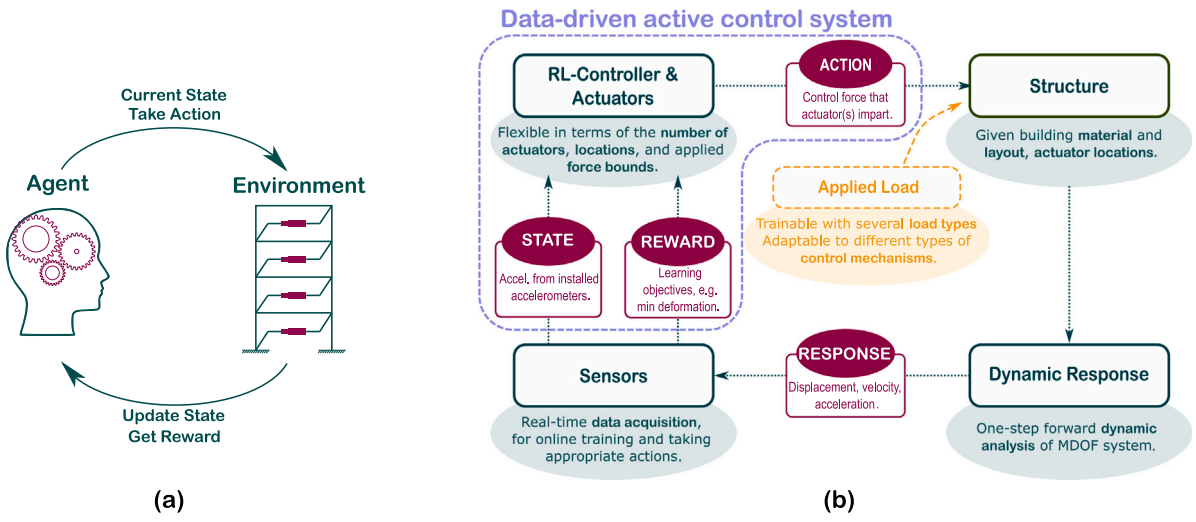


Fig. 1. (a) Standard RL framework. This involves the interaction of an agent with an environment, wherein the agent learns to take a set of actions depending on the state of the environment in order to maximize rewards which is defined based on a set of required performance criteria. (b) Active structural control as an RL problem. The structure of interest is the environment; the state is a subset of the full state–space of the dynamic response of the structure measured using the deployed sensor network; the reward is a function that helps achieving the goals of structural control, i.e., response minimization; the agent is the control system that regulates the actuators; the action is the control force that the actuators generate which in turn is imparted on the structure.

with optimization algorithms such as high variation in the objective function, and convergence to local minimum or saddle point to name a few. There have been many recent algorithmic developments that enhance the various optimization processes for both on- and off-policy methods that address some of these issues. The choice of algorithm employed thus becomes application-specific, such that an optimal policy is achieved with relatively lower computational effort.

Active structural control can be formulated as a deep RL problem. In these problems, the exogenous input is defined by external loads that are applied to the structure, such as loads induced by ground motions and wind. The structure is the environment, in the RL context, that reacts to the input and responds dynamically. A subspace of the full state–space of the structure's response is monitored by a sensor network (for example, a network of accelerometers) and will be used as the state vector in the RL framework. Note that the state for RL and state–space in the context of control theory are distinct and should not be perceived interchangeably. In addition, using the sensory measurements, the framework calculates the immediate reward for each discrete time step. The reward function incentivizes smaller deformation, vibrations, and actuator forces. Given the state vector, an active controller that is trained with RL, predicts the optimal action: the associated actuator forces for a given time. The actuator forces and the ground motion acceleration in the new time step are input to the structure for the new cycle. Fig. 1(b) shows how the active structural control problem is formulated as an RL problem, with comparisons drawn between the various components of a general RL paradigm and the structural control problem.

The RL framework used in this study enables off-policy and on-policy learning. For the on-policy setting, one desired functionality is an integrated structural dynamic simulation solely for the data generation purpose based on user-defined structural properties. The simulation framework is designed as a Gym environment as part of the open source *openAI* framework. The readers should note that the simulator does not contribute to the learning process and merely acts as a data simulation environment, as it is common in model-free RL settings.

In this paper, an exclusive Gym environment for structural control problems is developed. Based on the parallels between active structural control and RL, the various RL attributes translate into the Gym environment as a function of variables associated with structural control. The *state* in this simulation (Eq. (4)) consists of the acceleration of the instrumented levels, external load (in this paper, seismic ground acceleration), and the most recent applied actuator (control) forces. The choice of acceleration as a structural response for inclusion in the state stems from the relative ease and ubiquity of accelerometer deployment on structures. In general, this could be any sufficient subset of the full state–space of the structure that is being measured. The proposed state function is found to be sufficiently informative for guiding the RL agent to effectively control the system response.

$$\mathbf{s} = \{\ddot{\mathbf{x}}, \ddot{\mathbf{x}}^g, \mathbf{a}\} \tag{4}$$

where $\mathbf{s}$ represents the state, $\ddot{\mathbf{x}}$ is the acceleration vector, $\ddot{\mathbf{x}}^g$ is the ground acceleration and $\mathbf{a}$ is the action taken by the policy, all in the last time step.

The readers should note that the simulation component of the proposed Gym environment is not a necessity for the RL-Controller. It aids the training process for the numerical examples used in this paper. If field structural response data is available for a structure, one could directly use the data for training RL-Controller. Furthermore, for a given data set, RL-Controller does not perform an explicit system identification to establish a full state space response model for a structure. Instead, the underlying neural network learns the structural model in an abstract space. This makes the approach model-free in addition to being purely data-driven.

Eq. (5) shows the *reward* function for the RL class, which is a function (f^r) of displacements, base shear, and applied control force. Again, this may be modified according to the performance needs for a structure. In this study, the actions are the applied control forces on structure through the deployed actuators.

$$\mathbf{r} = f^r(\mathbf{x}, V_b, \mathbf{a}) \tag{5}$$

where $\mathbf{r}$ shows the reward value, $\mathbf{x}$ is the story displacement vector, V_b is the base shear. The arguments of reward function require minimization for efficient response control. Therefore, each term in the reward function is accompanied by a negative sign (e.g., minimizing displacement maximizes the reward). It should be noted that all variables are scaled in the reward function to promote numerical stability as further discussed in 4.3.

4.2. RL-controller

In this section, the proposed RL-based control system class, RL-Controller [76], its attributes, underlying methodology, and its various functionalities are introduced. The brief discussion is shown in Fig. 2.

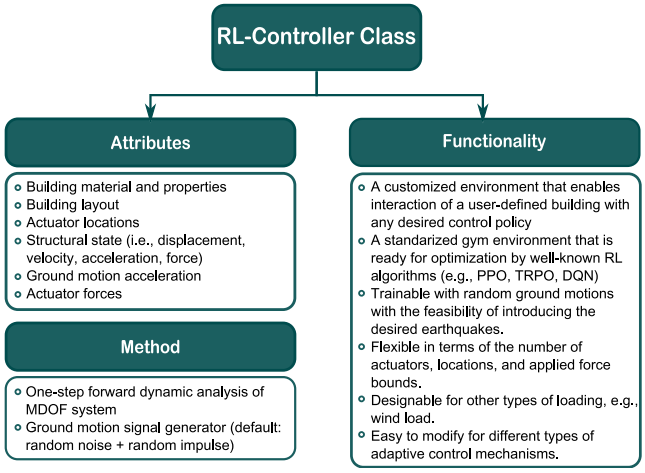


Fig. 2. Features of the proposed RL-Controller class developed as a Gym environment.

Attributes: The RL-Controller class takes structural properties, actuator layout, and the range of control forces as the input. To maintain a model-free controller, the user must measure structural responses such as story accelerations in this platform or has to construct a trustworthy model for response data generation for training. The state vector in RL setup can be defined based on the sensory devices that are available. To facilitate the learning of the inherent dynamics of the system, in addition to response at time t, responses at l previous time steps are also included. Consequently, the length of the structural response history affects the performance of the learning-based agent in estimating the control force. Hence, the number of time steps, l, is defined as a user-defined parameter in this framework. The class estimates control forces in real time for the structure based on the ground motion acceleration it is subjected to (Fig. 1(b)). These control forces act on the structure simultaneously with the external loads. Finally, depending on the task, users can define a time delay between the state and the action to take into account possible practical actuator delays.

Method: RL-Controller requires response data from the dynamic behavior of a structure. In this paper it is achieved through numerical simulation of the dynamic response of a multi degrees of freedom (MDOF) system using the Newmark-β method [77]. For the training process, the agent randomly explores the environment by applying control forces that each actuator can subject a structure to. This ensures that the controller captures the impact of the applied control force on the reward for each given ground motion and is prepared for unforeseen combinations of external load that the structure may encounter in the future. Furthermore, the structure is subjected to random loads during the training process. To enhance robustness, a combination of white noise and random impulses is used as simulated ground motions for training. This loading strategy is found to be simple, minimal, and effective when the trained structure is subjected to real earthquake records. This is because earthquake records typically consist of strong pulses that lead to the largest structural responses. The ground motion generator is in essence an external load generator and can be modified to account for characteristic features of other forms of loads. It should be emphasized that this data simulator is independent of the training process and the RL agent solely uses the generated data to find optimized policy.

Functionality: The proposed class RL-Controller is a customized Gym environment with flexible user-defined parameters. The defined class can be used for learning the optimal control policy using many novel optimization methods to meet the user's needs. The training process is designed to be efficient while it is capable of handling any customized structural demands, i.e., desired ground motions. Same as any Gym environment, the RL-Controller class is flexible to changes in attributes or optimization parameters to match the problem and action level the best. For example, a user may assign any sufficient subset or function of the full state–space as the RL state vector, or a user may simply modify the environment for a variety of loading regimes such as wind and wave loads. In addition, the class can be modified to incorporate specific control mechanisms such as active tendons, active tuned mass dampers, and active viscous dampers, to name a few.

4.3. Optimization and training

To select an efficient optimization technique in terms of rate of convergence and efficacy, several methods were compared, with proximal policy optimization (PPO) and soft-actor–critic (SAC) showing the best performance. Both methods are from the family of Actor–Critic RL methods. Actor–Critic methods are Temporal Difference (TD) reinforcement learning methods that allocate separate memory to track policy regardless of value function [75]. The *actor* or policy structure is responsible for taking actions while the value function or *critic* criticizes the actions (and the controlling policy) by tracing TD error. PPO is an on-policy gradient technique based on trust region method which uses an objective function with clipped probability ratios that forms a pessimistic estimate (i.e., lower bound) of the performance of the policy to avoid high variance in sample-based learning [78]. SAC is an off-policy optimization method which seek to maximize entropy for broader exploration, and the expected reward for policy improvement at the same time (see Eq. (6)) [79]. These two methods (SAC and PPO) work effectively in this control problem and both reach reasonable policies in each training event. PPO can gain a stable improved policy quickly in most training sessions, although, SAC is found to be a better choice in this problem due to its robustness and more extensive action-space exploration. Consequently, all the discussions in Section 5 are based on SAC as the optimization method for finding the optimal policy. SAC [79] attempts to find a policy as shown in Eq. (6) that maximizes the entropy objective:

$$\pi^* = \arg\max_{x} \sum_{t=0}^{T} \mathbb{E}_{(s_t,a_t)\sim\tau_\pi}[\gamma^t(r(s_t,a_t)) + \alpha\mathcal{H}(\pi(.|s_t))] \tag{6}$$

where, π and π^* are a given policy and optimal policy respectively. T is the number of time steps and $r : S \times A \to \mathcal{R}$ is the reward function. $\gamma^t \in [0,1]$ is the discount rate at time t to ensure that the sum of expected rewards and entropies is finite, $s_t \in S$ and $a_t \in A$ are the state and action at time step t, τ_π is the distribution of trajectories induced by policy π, α determines the relative importance of the entropy term versus the reward known as the temperature parameter, and $\mathcal{H}(\pi(.|s_t))$ is the entropy of the policy π at state s_t and is calculated as $\mathcal{H}(\pi(.|s_t)) = -\log(\pi(.|s_t))$. To maximize the objective, SAC uses soft policy iteration which is a method of alternating between policy evaluation and policy improvement within the maximum entropy framework. The policy evaluation step involves computing the value of policy π. To do this the soft state value function is defined as:

$$V(s_t) := \mathbb{E}_{a_t\sim\pi}[Q(s_t,a_t) - \alpha\log(\pi(a_t|s_t))] \tag{7}$$

In the policy improvement step, function $Q_\theta(s_t,a_t)$ – the approximated action-state value function – is parameterized by θ (commonly modeled as a neural network) and the objective function in Eq. (8) is optimized in order to minimize the soft Bellman residual:

$$J_Q(\theta) = \mathbb{E}_{(s_t,a_t)\sim D}[\frac{1}{2}(Q_\theta(s_t,a_t) - (r(s_t,a_t) + \gamma\mathbb{E}_{s_{t+1}\sim p(s_t,a_t)}[V_{\bar{\theta}}(s_{t+1})]))^2] \tag{8}$$

here D is a replay buffer of past experiences and $V_{\bar{\theta}}(s_{t+1})$ is the estimated value at state s_{t+1} using a target network for Q and a Monte-Carlo estimate of Eq. (7) after sampling experiences from the replay buffer. The policy improvement involves updating the policy in the direction with maximum gained rewards. To do this, the soft Q-function is calculated in the policy evaluation step to track policy changes. Specifically, the policy is updated towards the exponential of the new

soft Q-function. After updating the policy towards the exponential of the soft Q-function then it will be projected back into the space of acceptable policies using the information projection defined in terms of Kullback–Leibler (KL) divergence. The policy parameters are learned by minimizing the expected KL-divergence with the objective function shown in Eq. (9) [80]. Note that here the policy network is parameterized by ϕ:

$$J_\pi(\phi) = \mathbb{E}_{s_t \sim D, a_t \sim \pi_\phi}[\alpha log(\pi_\phi(a_t|s_t)) - Q_\theta(s_t, a_t)] \tag{9}$$

As both objective functions are parameterized and differentiable, the parameters can be updated by error back-propagation using conventional stochastic gradient ascent based approaches. In this study, the policy and value functions are modeled as multi-layer perceptrons with three hidden layers, each of size 128. More details of this network can be found in Appendix A. Note that being a standard Gym environment, RL-Controller provides this flexibility to experiment with various state-of-the-art optimization algorithms and network architectures with minimal effort.

To train the RL-Controller, first the data simulation and reward functions must be formulated and introduced to start the learning process. Throughout the training process, the action or actuator forces are estimated based on the most recent updated policy (random at first). The state and action along with the given ground motion are inputs to update the state using the function f^{sim} of data simulator and obtain reward of this action from the f^r. In the end, the policy is updated using the SAC algorithm. By repeating this training process for sufficient numbers of iterations, the optimizer is able to find a reliable optimal control strategy. Algorithm 1 shows the details of the training process. The subscripts in the algorithm shows the time step, π_θ is the policy, and s, a are the temporal state and action vectors. $x, \dot{x}, \ddot{x}$ show the instantaneous story displacement, velocity, and acceleration, $\ddot{x}^g$ is the immediate ground motion acceleration. N and T are respectively number of iterations and length of each episode. l is the length of response history included in the state, and SAC represents the optimization step taken for each training iteration.

Algorithm 1 RL-Controller training process with simulator

1: Introducing f^{sim}, f^r.
2: $s_0 = \{\ddot{x}_0, \ddot{x}_0^g, a_0\}$.
3: **for** *iteration* $= 1, 2, \dots, N$ **do**
4: **for** $t = 0, \Delta t, 2\Delta t, \dots, T$ **do**
5: $a_t \leftarrow \pi_\theta(s_t)$. ▷ Choosing action for the current state according to policy π_θ
6: $x_{t+1}, \dot{x}_{t+1}, \ddot{x}_{t+1} \leftarrow f^{sim}(x_t, \dot{x}_t, \ddot{x}_t, \ddot{x}_t^g, a_t)$. ▷ Finding next state of structure given action a_t
7: $s_{t+1} = \{\ddot{x}_{t-l:t+1}, \ddot{x}^g_{t-l:t+1}, a_{t-l:t+1}\}$. ▷ Defining the state for the next time step
8: $r(s_t, a_t) \leftarrow f^r(x_t, M\ddot{x}_t, a_t)$. ▷ Calculating reward for taken action in previous state
9: $\pi_\theta \leftarrow SAC(\pi_\theta, \{s_t, r(s_t, a_t), s_{t+1}\})$. ▷ Updating policy π_θ based on the latest reward

In this RL-based controller design, it is critical to design a reward function that can efficiently guide the optimization process. The designed composite reward function consists of four terms as defined in Eq. (10): (1) quantifier of total inter-story drifts P_1, (2) quantifier of total base shear P_2, (3) the absolute displacement of the first story P_3, and (4) the total sum of action forces, i.e., the total applied control force P_4. The scales β_2, β_4 in reward components are chosen based on physical scale of each component for the studied structure to keep reward maximization effective for all four desired properties along training. In Eq. (10), ISD represents interstory drift, V_b is base shear, Δt is time step, $x_{:,1}$ is first story displacement, u is the applied force, and n_{DOF} is number of degrees of freedom. In order to combine these terms into one scalar quantity as reward, various compositions can be defined. In case of using a weighted sum of the terms with constant factors, the terms will improve unevenly. It is also nontrivial to find a set of proper static weights. More investigation on this comparison is shown in Fig. 4 and discussed in the following. To avoid that, a weighted sum composition with adaptive weights is introduced to facilitate the improvements between the four terms uniformly. Fig. 3 presents a comparison between the two approaches. Note that the term *Loss* in this figure is equal to the reward function times minus one. The adaptive weighting terms $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ are calculated as follows:

$$P_1 = (\sum_{i=1}^{n_{DOF}} |ISD|)^2, P_2 = \beta_2 V_b (\Delta t)^2, P_3 = |x_{:,1}|, P_4 = \beta_4 \sum_{i=1}^{n_{DOF}} |u| \tag{10}$$

$$r = -\sum_{i=1}^{4} \alpha_i P_i \tag{11}$$

$$\alpha_i = \frac{P_i}{\sum_{j=1}^{4} P_j} \tag{12}$$

Fig. 3 shows that when constant weights are used to construct the loss function, the loss terms decay non-uniformly. Alternatively, when the terms are combined using adaptive weights, the majority of the loss terms (except P_1) become synchronous and follow the same decaying regime. This is advantageous for reaching a more optimal solution wherein conflicting variables are involved. For example, a large reduction in deformations will contribute towards a high reward, however, it will be offset by a large control force. Instead, if all the performance parameters reduce in tandem, one would reach a better compromise between the various conflicting variables.

Fig. 4 compares the distribution of the maximum response of the five story building (from Section 5.2) subjected to the seven selected earthquakes as mentioned in Table 1 for optimal models trained once with constant weights for the reward terms and once using the proposed adaptive weights. The 95% confidence interval (CI) distribution and mean value of responses are achieved to be less in most cases with the adaptive weights. Furthermore, it can be observed that reward terms associated with structural response are lower for the adaptive weights compared to the constant weights, namely, ISD in the first story (P_3), sum of the squares of maximum ISDs for each story (P_1), and base shear which is a linear combination of the accelerations weighted with the respective story masses (P_2). Hence, the efficacy of adaptive weights is clear.

5. Results

This section presents three numerical studies that evaluate the performance of the proposed RL-Controller class. The scalability of the RL solutions for continuous state and action spaces such as control problems is known to be challenging [81]. Therefore, in this section three case studies that are distinctive in terms of the number of actuators are evaluated. The first case study is on a three story frame structure as a proof of concept for the proposed methodology; the second case study is a five story shear building that compares the performance of the proposed framework to a traditional optimal control algorithm; and the last case study demonstrates the efficacy of the proposed framework in controlling structures exposed to extreme loads wherein nonlinear behavior is expected. It should be noted that the structures in the first two cases studies are assumed to behave linearly. For all case studies, RL-Controller was trained using a combination of white noise and random impulse as a ground motion as shown in Appendix B. Subsequently, the performance is evaluated for seven different ground motions. These ground motions have a variety of amplitudes, frequency content, and duration, thus capturing the uncertainties associated to ground motion parameters. Table 1 lists of the selected ground motions and their peak ground accelerations (PGA).

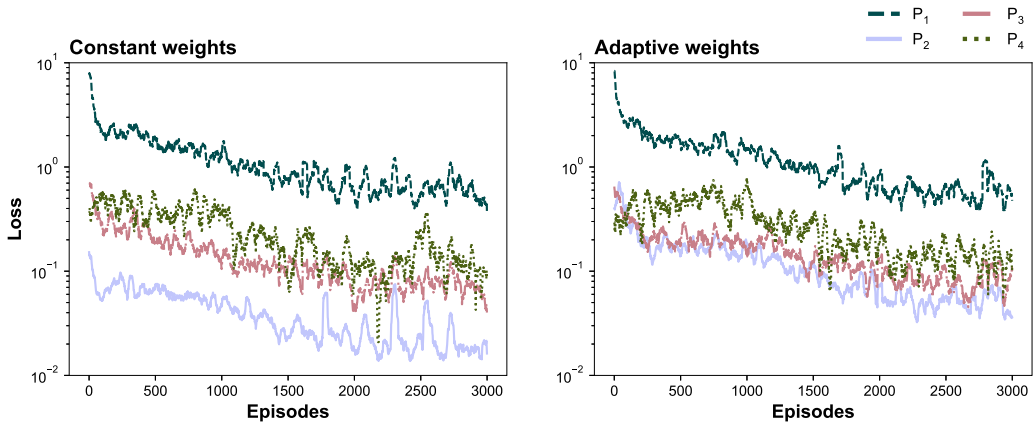


Fig. 3. Learning curve comparison: when constant weights are used in the loss function (mirrored version of the reward), the reduction of the losses as well as their values is not uniform. In contrast, by using adaptive weights, the loss values in three components (expect P_1) become similar.

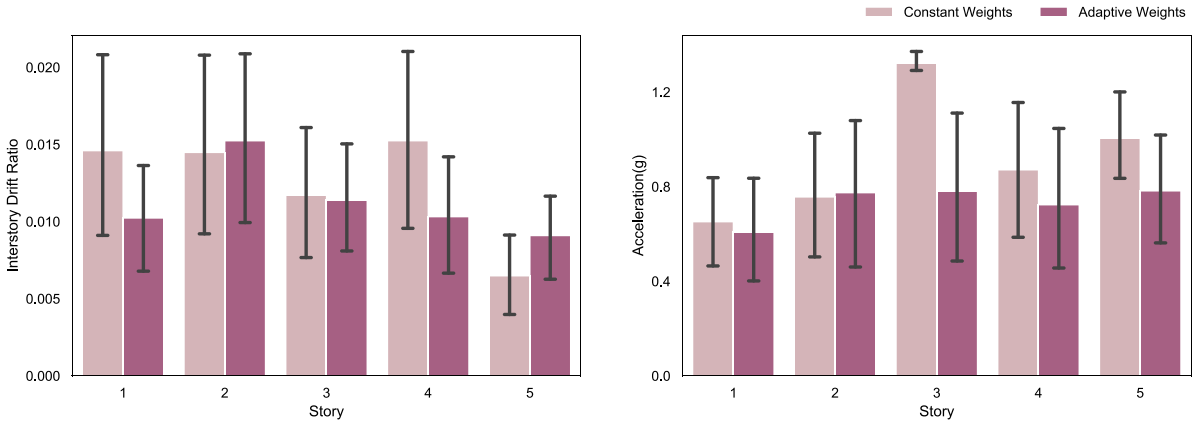


Fig. 4. Response distribution (95% CI) comparison of two optimal models subjected to seven earthquakes: Model with constant weights in reward function, and model using adaptive weights in reward function terms.

Table 1
Earthquake records [82] used for the evaluation purposes.

Name	Earthquake record and location	Year	PGA (g)
Loma Prieta	Loma Prieta, Alamaeda Naval Air Station Hangar	1989	0.27
Imperial Valley	Imperial Valley-06, El Centro #4	1979	0.48
Coalinga	Coalinga-05, Oil City Road	1983	0.86
Kobe	Kobe, Takatori	1995	0.62
Chi Chi	Chi Chi CHY101	1999	0.35
Sylmar	Northridge-01, Sylmar Converter Station	1994	0.61
W Pico	Northridge-01, West Pico Canyon Road	1994	0.46

5.1. Case study one: Three story frame structure

The first case study is a three story frame structure modeled as an equivalent linear three degree of freedom system based on the experimental setup from Chung et al. [83]. The system matrices (mass, damping and stiffness) for this structure are as follows:

$$\mathbf{M} = \begin{bmatrix} 1002.4 & 0 & 0 \\ 0 & 1002.4 & 0 \\ 0 & 0 & 1002.4 \end{bmatrix} \text{kg},$$

$$\mathbf{C} = \begin{bmatrix} 391.12 & -58.53 & 63.01 \\ -58.53 & 466.83 & -0.27 \\ 63.01 & -0.27 & 446.97 \end{bmatrix} \text{N-s/m},$$

$$\mathbf{K} = \begin{bmatrix} 2.80 & -1.68 & 0.38 \\ -1.68 & 3.09 & -1.66 \\ 0.38 & -1.66 & 1.36 \end{bmatrix} \times 10^6 \text{ N/m} \quad (13)$$

One actuator located at the third story is considered for this case study. Acceleration is measured at each story at a sampling rate of 50 Hz. RL-Controller class for this benchmark building trains the RL agent for three million iterations to reach optimal policy using SAC. During training, an efficient performance is reached with hyperparameter $l = 5$, and learning rate 3×10^{-4}. Furthermore, to enhance the impact of actions over a longer period of time, the discount factor γ is set to 0.999 for weighing temporally distant rewards.

Fig. 5 compares the inter-story drifts (ISD), accelerations, and story shears at each story between the uncontrolled system and the controlled system deployed with RL-Controller when the structure is subjected to the Coalinga (1983) earthquake. Significant reduction appears in structural response in most cases, e.g. 92% and 70% reductions in average ISD and acceleration values for the third story, respectively. In addition to lower amplitudes, the control system is able to reduce the number of moderate vibration cycles after the main shock in the structure thus enhancing the fatigue life of the structure. In the case of accelerations however, a relatively large peak amplitude is observed. This is due to the contribution from the large control force that is applied on the structure that enabled a significant reduction in the ISDs. Since, larger control forces lead to lower deformation but higher accelerations, there is a trade off between these two performance measures as defined by the designer in the reward function. Fig. 5 also shows the story force–deformation characteristics comparing the controlled and uncontrolled systems. In addition to reduction in story shear and deformations, an apparent increased stiffness (larger slope) and damping (larger elliptical shape of the curve) of the structure can be observed which facilitates the reductions in deformation and story

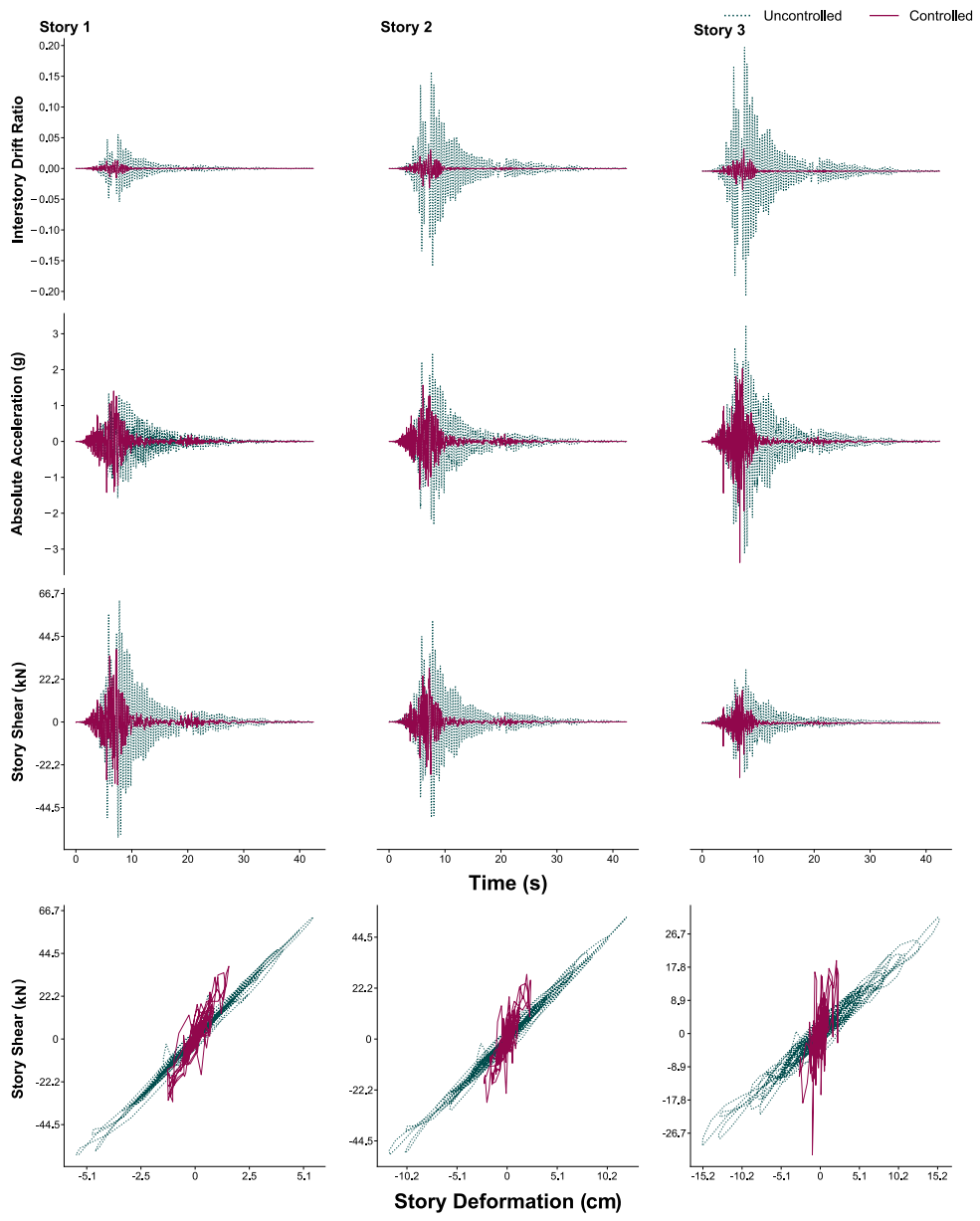


Fig. 5. Case study one: Three story frame structure. Comparison between ISD, accelerations, story shears, and force-deformation curves for each story between uncontrolled structure and the structure when deployed with one actuator driven by RL-Controller. The structure is subjected to unscaled Coalinga (1983) earthquake.

shears. This shows that RL-Controller inherently learns to influence structures, imitating traditional controllers by modifying structural properties, i.e. increase in stiffness and damping, in order to minimize structural response purely from data.

5.2. Case study two: Five story building

For the second case study, a five story shear building is modeled as a linear MDOF system as described in Park and Noh [84]. Table 2 shows the structural properties of each degree of freedom. The damping is modeled as Rayleigh damping such that damping ratios are 1% and 5% at first and fifth modes. Acceleration is measured at every story at a sampling rate of 100 Hz. Three different actuator setups are discussed: (a) actuators at stories two and four (b) actuators at stories one, three and five, and (c) actuators at all stories. The location of the actuators in this case study are selected arbitrarily and did not involve an optimal actuator placement study.

The RL-Controller class trains the RL agent in this benchmark building for 1.5 million iterations to reach optimal policy with SAC. The optimal performance is found when hyperparameter $l = 5$ is set, learning rate is 1.5×10^{-4}, and discount factor is $\gamma = 0.999$. In this section, the comparison is between the performance of RL-Controller and a traditional optimal control algorithm, LQG. To obtain a fair comparison between the two algorithms, the same control variables are used for LQG, namely, ISD, base shear and the control force. As discussed earlier, a novel feature of the proposed framework is the use of adaptive weights in the reward function of RL-Controller. However, traditional LQG uses constant weights throughout. For a fair comparison, the weight matrices $\mathbf{Q}$ and $\mathbf{R}$ in LQG were defined such that they are equal to the average scale factors over the training process of the RL agent.

In order to quantify performance, eight metrics J_1 to J_8 are defined as shown in Table 3. In the metrics' definition, subscripts C and UC represent controlled and uncontrolled response. δ, $\dot{x}$ and $\ddot{x}$ are the ISD, story velocity and acceleration, respectively. Furthermore, u is the control force imparted on the structure. V and $V_{b,.}$ represent the story shear and base shear, respectively. E in equations J_3, J_6, J_7 and J_8 is the signal energy as defined in Eq. (14).

Table 2
Five story building model properties.

Story	One	Two	Three	Four	Five
Mass ($\times 10^3$ kg)	25	20	20	18	15
Stiffness ($\times 10^6$ N/m)	5	4	4	3	3

Table 3
Metrics used to evaluate the performance of the controllers.

Metric	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8
Def.	$\frac{\max \lvert\delta_C\rvert}{\max \lvert\delta_{UC}\rvert}$	$\frac{\max \lvert\ddot{x}_C\rvert}{\max \lvert\ddot{x}_{UC}\rvert}$	$\frac{E_u}{E_{V_{b,UC}}}$	$\frac{\max \lvert u\rvert}{\max \lvert V_{b,UC}\rvert}$	$\frac{\max \lvert V_C\rvert}{\max \lvert V_{UC}\rvert}$	$\frac{E_{\delta_C}}{E_{\delta_{UC}}}$	$\frac{E_{\dot{x}_C}}{E_{\dot{x}_{UC}}}$	$\frac{E_{\ddot{x}_C}}{E_{\ddot{x}_{UC}}}$

$$E_x = \int_0^T |x(t)|^2 \, dt \tag{14}$$

Fig. 6 compares J_1, J_2, and J_5 between RL-Controller and LQG for the aforementioned three actuator placements. The bold lines in each figure represents the average J value over all the seven selected ground motions. The dashed lines are the J values obtained from structural response to individual earthquakes (from Table 1). Clearly, RL-Controller outperforms LQG on average considering the ISD, story acceleration and story shear. For example, in case (b) in average, 25% lower ISD ratio, 25% lower acceleration ratio and 34% lower story shear ratio are achieved with RL-Controller compared to LQG. In addition to analysis of the story-specific performance metrics defined earlier, it is important to compare the base shear and the total control force required to minimize structural response. In order to do so, two additional metrics are used as defined in Eqs. (15) and (16).

$$H_1 = \frac{\max |V_{b,C}|}{\max |V_{b,UC}|} \tag{15}$$

H_1 shows the ratio between maximum controlled and uncontrolled base shear.

$$H_2 = \frac{\max \sum u}{\max |V_{b,UC}|} \tag{16}$$

H_2 shows the comparison for total applied load scaled to uncontrolled base shear.

Fig. 7 shows a box plot for H_1 and H_2. It is clear that for all the actuator deployment schemes considered, RL-Controller yields lower base shears as well as lower control forces. Additionally, the standard deviation of the metrics for RL-controller is lower than LQG. This implies that RL-Controller performs better with unknown loads. It is believed that one reason for this enhanced performance for unknown loads is the use of a combination of white noise and random impulse during training. LQG however, is limited to a white noise excitation assumption. Clearly, taking into account the characteristics of ground motions with a simple random impulse for training enhanced the performance.

By comparing the median values for H_1 and H_2 for RL-Controller, a nonlinear trend can be observed. For H_2 it is expected that when there are more actuators, the total control force will be significantly higher. However, there is a slight reduction in the total control force when three actuators are deployed compared to two. This is also reflected in the median values of H_1. This is further reinforced by Fig. 6, where clear improvements for RL-controller in J_1, J_2, and J_5 is evident, going from the two actuator to the three actuator configuration. However, moving from three to five actuators there is no substantial change in the overall performance. In fact, in the case of story shears, the three actuator deployment scheme performs better than a case where actuators are deployed on every story. This demonstrates that for a given structure there might be certain combination of actuator number and location that might be optimal. It should be noted that here to make a fair and tractable comparison between different actuator configurations, we optimize the RL-Controller using a certain number of iterations. Although, in a possible global optimum for these configurations, the case with five actuator deployment at worst is expected to learn to act like a three actuator setup instead of delivering inferior performance.

Based on the results so far, it is clear that RL-Controller outperforms LQG on average for all the ground motions considered. It is discernible that RL-Controller produces lower control forces compared to LQG. This would imply a potentially lower base and story shears. To elaborate, a comparison of applied force in the actuator setup (b) subjected to W Pico earthquake (Table 1) among RL and LQG is shown in Fig. 8 to compare the optimal control strategies. As mentioned, it can be seen that LQG is imposing higher noise in the applied control force compared to RL which causes RL's enhanced performance. However, it is counter-intuitive to observe that the ISDs are also lower for RL-Controller even with lower maximum control forces. Since, RL-Controller produces a black-box agent, it is a challenge to interpret the internal dynamics of the trained model. However, intuitively the possible reasons for an improved performance are described next.

There are two potential sources that lead to this enhanced performance. First, during the training process, RL-Controller was trained using a combination of white noise and random impulse loads. LQG on the other hand assumes a Gaussian white noise excitation. This indicates that RL was better equipped to handle unknown ground motions, as ground motions are typically comprised of broadband signals and a few strong pulses. Further details of the proposed ground motion signal generator is shown in Appendix B.

Second, the adaptive weights in RL-Controller's designed reward function enhances the performance compared to LQG wherein the weights associated to the minimization problem are constant. Therefore, if a larger ISD occurs in the structure at a certain instant of time, RL-Controller adaptively penalizes displacement more than the rest of the terms of reward function in the subsequent time step, while LQG does not adjust the policy at all. This might lead to greater reductions in ISD when the same control force is applied. Finally, note that the RL-Controller – a model-free agent – is outperforming LQG, which is a model-aware controller. This is a unique and promising advantage of the proposed data-driven approach.

5.3. Case study three: Nonlinear five-story building

For the third case study, the structure from the second case study is reused but with a elasto-plastic nonlinear story behavior for each of the five stories. This is modeled such that once the story force exceeds the yield force, the columns in that story show perfectly plastic behavior. The yield story forces for each of the five stories in ton-force are as follows 37.5, 45.7, 45.7, 26.9, 26.9. In this study, one actuator is located at each story to control the structure by five total actuators. Through this case study we demonstrate the efficacy of RL-Controller when dealing with a nonlinear control case.

The RL-controller is trained exactly in the same way as it was for the linear cases, i.e., with external loads modeled as a combination of a Gaussian white noise and impulse forces, and a reward function wherein each parameter is adaptively weighted. Subsequently, the structure is tested when subjected to strong ground motions that lead to nonlinear behavior in the structure.

Fig. 9 compares the response at the first story of the uncontrolled versus RL-controller controlled structures when subjected to three different ground motions, namely, Chi Chi (1999), Kobe (1995) and Sylmar (1994) scaled up by a factor of 1.5 to adjust to the training scale. It is clear that for the uncontrolled structure, nonlinear excursions are observed due to yielding and subsequent plastic deformations. RL-controller effectively controls the drift response and limits the structural yielding. This observation is further reinforced by the story force deformation curves for each ground motion. RL-controller has simultaneously reduced both the deformation and the story forces experienced by the structure.

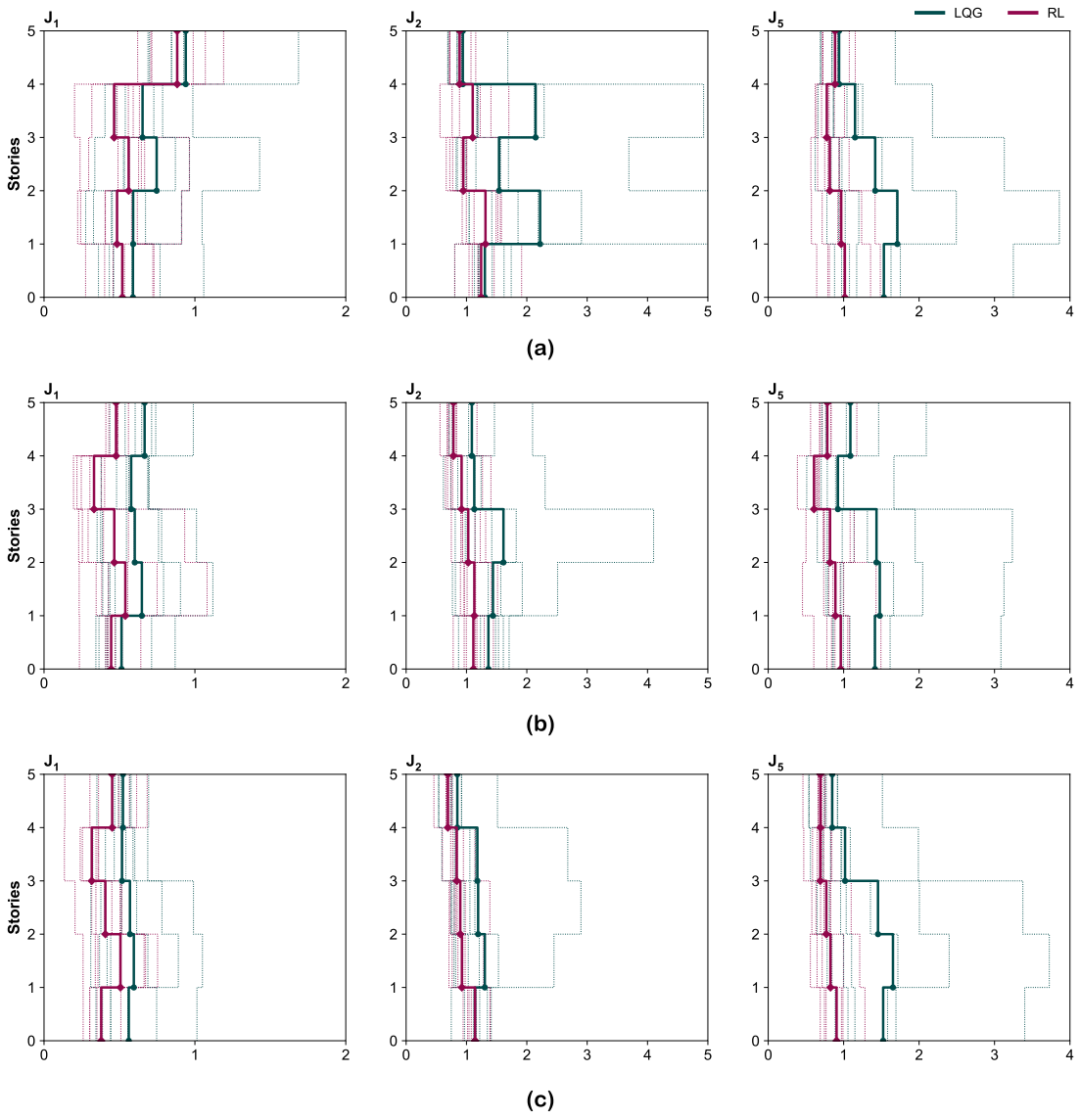


Fig. 6. Case study two: Five story building. Comparison of J_1, J_2, and J_5 between RL-Controller and LQG for all stories and three different actuator deployments: actuators at (a) stories two and four (b) stories one,three and five (c) all stories. The bold lines represent the average J values over all the seven earthquakes. The dotted lines are for each individual earthquake.

Table 4
Case study three: Performance metrics for the nonlinear structure averaged over seven ground motions.

Stories	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8
One	0.1917	0.6855	0.0481	0.0877	0.5348	0.1838	0.0679	4.331
Two	0.3202	0.7279	0.0317	0.0882	0.5767	0.1320	0.0459	1.6937
Three	0.3918	0.7681	0.0122	0.0862	0.5500	0.1023	0.0524	0.4735
Four	0.3535	0.6854	0.0275	0.0884	0.5904	0.2857	0.0347	0.7903
Five	0.5061	0.6028	0.0124	0.0807	0.6028	0.2457	0.0350	0.3506

Table 4 shows the various performance metrics J_is (Table 3) averaged over the seven ground motions given (Table 1) for each story of the structure to demonstrate the overall performance of RL-Controller. The values of J_1 demonstrate that the peak drift is effectively controlled. The increasing value of J_1 along the story height alludes to the typical decreasing nature of ISDs when the structure does not have a soft story. In addition to reduction in drifts, the peak story accelerations are also reduced, as can be noted from the values of J_2. J_3 being the measure of control force energy demonstrates that effort required by each actuator decreases with height. This is in agreement with typical controller behavior in structures. As was the case with J_2, J_5 demonstrates that the base shear is significantly reduced by RL-Controller. J_6, J_7 and J_8 quantify the energy in the response time histories of ISD, velocity and accelerations. As was demonstrated in Fig. 9, the ISDs are significantly reduced and the plastic deformations are controlled, leading to reductions in the ISD signal energy. A similar behavior is also observed for the velocity response energy. For the accelerations however, we observe an increase in the first story. Although the peak acceleration response is reduced as evidenced by J_2, the total energy increases due to RL-Controller being actively engaged in applying control forces to the structure to minimize the ISD.

6. Considerations for a practical implementation

The previous sections demonstrate the efficacy of the proposed RL-based active control framework, RL-Controller, through three numerical case studies. The proposed controller leads to significant reductions in structural response quantities when subjected to exogenous loads. Furthermore it outperforms LQG controller, a traditional optimal control algorithm.

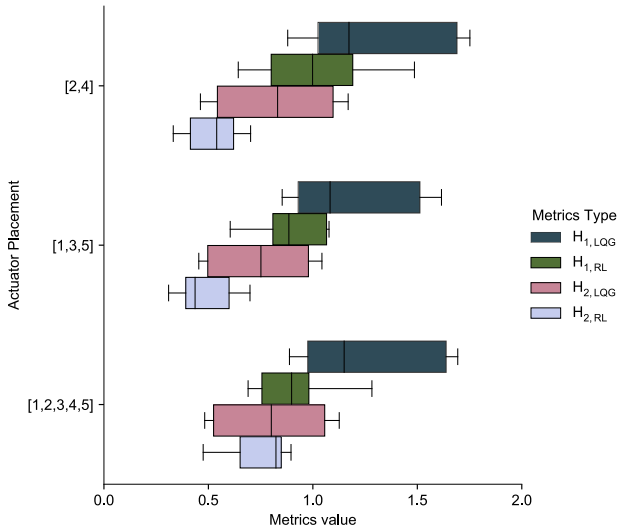


Fig. 7. Case study two: Five story building. Comparison of H_1 and H_2 between RL-Controller and LQG for seven earthquake records. In addition to performance comparison, this also shows the variation due to actuator deployment.

This section focuses on concerns associated with the implementation of the control system. In particular, first is a study of the variations in the structural properties and its impact on the performance of the trained RL-Controller. The possible degradation or retrofit will normally change the structural properties that the control system is unaccustomed to; thus it is critical for the control system to perform robustly despite these uncertainties [85]. Next is an investigation of the inevitable time delay of the RL-Controller in applying forces and an evaluation of its performance [86,87]. In order to test these implementation issues, the three story building from the first case study is considered.

6.1. System property variations

Control systems typically are designed such that they are tuned to a set of system parameters. Among them, structural material and geometry play a significant role. For example, in the case of optimal control strategies, the control force is determined by a gain matrix that is a function of all system matrices. This is true for RL-Controller as well. Although, it does not explicitly require a model of the system, i.e. system matrices, it is still trained according to structural responses resulting from a certain unknown set of structural parameters.

Over the design life of a structure, its material and geometric properties gradually change due to aging. This leads to a situation where a control system may not be *tuned* to the structure. This section studies the impact of changes in structural properties on the performance of RL-Controller. This is done by using the trained model with the original structural system matrices for testing on the structure where the system matrices are varied. This analysis is achieved by individually varying the mass, damping, and stiffness matrices to study the impact of each on the performance of RL-Controller. Table 5 shows the performance metrics defined in Table 3 under 5% and 10% variation in the mass, stiffness and damping matrices. Note that the Js listed here are averaged over all seven ground motions listed in Table 1.

The variation of mass impacts the performance of RL-Controller more considerably, unlike stiffness and damping variation. For 10% decrease, most of the performance metrics increase compared to 5% and some surpass one (implying worse performance compared to the uncontrolled case). Although, increase in mass when up to 10%, does not adversely affect RL-Controller's performance (values are considerably lower than one). A significant reduction in mass will substantially

Table 5
Performance of RL-Controller when structural system matrices are varied. The performance is quantified using the metrics defined in Table 3. The J values are averaged over all seven ground motions.

Property	Var.(%)	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8
Mass	+10	0.2433	0.8966	0.1416	0.5583	0.9158	0.0346	0.0299	0.4250
	+5	0.2414	0.9353	0.1184	0.5640	0.9463	0.0342	0.0301	0.3787
	−5	0.2750	1.0777	0.1324	0.5731	0.9457	0.0386	0.0415	0.4561
	−10	0.2870	1.4144	1.5950	0.6871	1.1848	0.1162	0.2286	5.6835
Stiffness	+10	0.2251	0.9629	0.1279	0.5594	0.9249	0.0347	0.0334	0.4014
	+5	0.2259	0.9280	0.1213	0.5524	0.8871	0.0341	0.0316	0.4033
	−5	0.2518	0.9235	0.1223	0.5739	0.9112	0.0357	0.0337	0.3829
	−10	0.2443	0.8672	0.1167	0.5482	0.8442	0.0354	0.0334	0.3700
Damping	+10	0.2482	0.9589	0.1306	0.5647	0.9139	0.0385	0.0351	0.4329
	+5	0.2484	0.9502	0.1265	0.5614	0.9070	0.0374	0.0339	0.4201
	−5	0.2439	0.9325	0.1170	0.5550	0.8959	0.0348	0.0317	0.3913
	−10	0.2446	0.9254	0.1137	0.5515	0.8902	0.0337	0.0301	0.3806
Original structure		0.2487	0.9460	0.1233	0.5581	0.9034	0.0364	0.0330	0.4100

increase the modal frequencies of the structure for the same stiffness. The RL-Controller is however, not trained to control a system with higher natural frequencies. Due to the lower mass, sudden changes in the ground motion will take a shorter time to manifest its full impact on the structural response. Furthermore, a higher frequency of vibration implies that it takes a shorter time for the structure to move away from the motion peaks to its equilibrium position. The RL-Controller is not trained for these conditions and simply applies high control forces adapted to the original (trained) lower natural frequency to minimize the structural response in a slower pace. As a consequence of the low-frequency control force and energy, the peak accelerations, peak story shear and all the structural response energies increase.

The variation of stiffness does not lead to significant changes. RL-Controller is effective for up to 10% variation in the stiffness matrix. For the higher stiffness in the test structure, when RL-Controller encounters a certain system response, it still imparts control forces consistent with a system of the training structure. However regardless of the control force, the stiffer building absorbs more energy, and consequently leads to larger accelerations and base shears. This explains the general downward trends in J_2 to J_5 for decrease in stiffness. Overall decrease in J_2 is an evidence that the absorbed energy is lower when the structure is more flexible.

Variations in the damping matrix also does not significantly affect the performance of RL-Controller. Even with 10% variations in the damping matrix, no substantial change occurs in most of the metrics and RL-Controller is fully effective. Hence, RL-Controller is robust to variations in damping and stiffness up to 10% and for mass up to 5%.

6.2. Time delay

A control system consists of multiple electrical circuits that transfer sensory data to a controller, which in turn, transmits back a control force that is imparted on a structure of interest. The different components that constitute this chain of information can induce time delays into a control system. If unaccounted, such delays may lead to sub-optimal control performance and may even compromise the safety of the structure. There has been an extensive effort made by the research community to account for such delays in the context of traditional control. However, most of those approaches involve using the system matrices. Since RL-Controller is model-free, an alternative approach is necessary to address this issue.

In this analysis, a prior knowledge on the extent of time delay for the installed control system is considered. Under this assumption, an RL agent is trained considering random time delays at each time step. For the sake of simplicity and to demonstrate the efficacy of the approach, the training of random delay is limited such that, the possible time

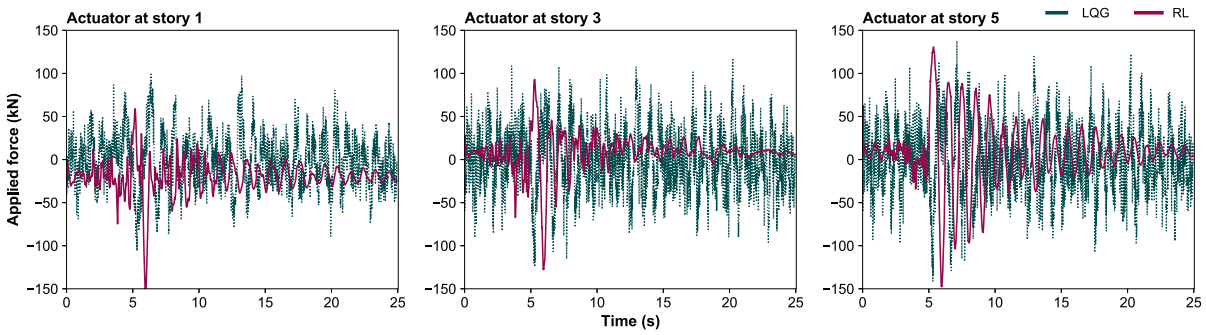


Fig. 8. Case study two: RL and LQG control force comparison subjected to W Pico earthquake.

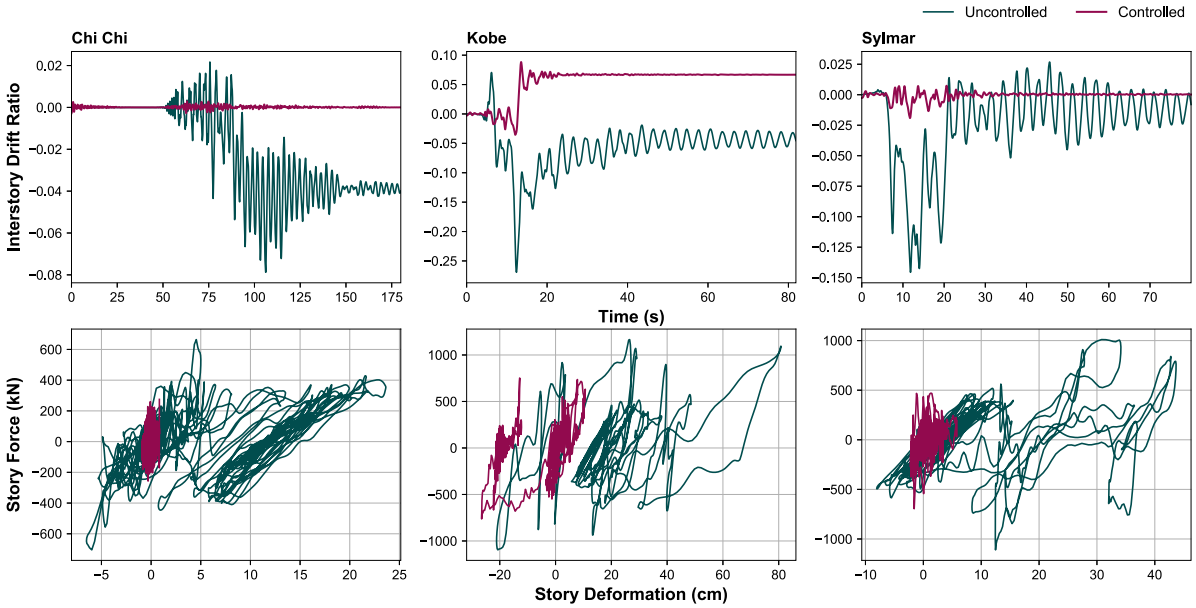


Fig. 9. Case study three: Nonlinear five-story building. Comparison between ISD and force–deformation curves for first story between uncontrolled and controlled structure when subjected to Chi Chi (1999), Kobe (1995) and Sylmar (1994) earthquake.

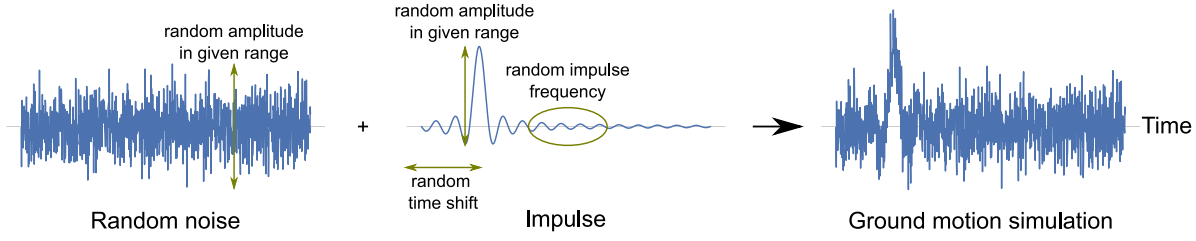


Fig. B.10. Schematic of the proposed ground motion signal generator including random noise and random impulse.

delays can be either no delay, one time step delay or two time steps delay (in this study, time step is 0.02 s). The space of the time delay random variable can easily be extended for a more comprehensive analysis based on the expected time delay in the system. This trained RL-Controller agent is then tested with three different cases: (a) there is a constant one time step time delay, (b) there is a constant two time step time delay, and (c) a random time delay (randomness restricted to no delay, one time step delay and two time step delay).

Table 6 shows the performance metrics for the newly trained RL-Controller with a random time delay. The performance metrics shown are the averaged values over all the seven ground motions considered in this study. It is clear that the trained network, although trained on a random time delay performs extremely well for a constant one time step delay. However, a two time step delay adversely affects the performance of RL-Controller. This implies that the actual time delay of the system should not be near the upper bound of the delays used to train the network. If however, the actual time delay is close to the mean

Table 6
Impact of time delay on the performance metrics for RL-Controller. The metrics are averaged over all seven ground motions.

Delay type	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8
One-step delay	0.2743	0.9357	0.0985	0.5217	0.7974	0.0393	0.0001	0.3534
Two-step delay	0.4651	1.9374	0.1039	0.6604	1.4403	0.1041	0.0006	1.7804
Random delay	0.3760	2.6489	0.1464	0.7048	1.2598	0.0622	0.0007	2.8714

or the median of the distribution, a robust performance in tackling the time delay issue is anticipated from RL-Controller. Note that, even in the worst case scenario (two-step delay), the performance metrics mostly confirm the efficacy.

7. Conclusion

This paper presents deep reinforcement learning as an adaptive and model-free solution to active structural control formulated as an MDP problem. In particular, a standard Gym environment is developed to define, experiment, and modify control systems named as RL-Controller which yields an optimal real-time control force strategy when trained. RL-Controller generates optimal policies by minimizing structural responses such as story deformation and accelerations, while simultaneously attempting to minimize the applied control forces. This paper demonstrates the efficacy of the proposed controller through three numerical case studies wherein the structures are subjected to a set of ground motions. First one demonstrates the response control ability of the proposed framework for a three story frame structure. Up to 92% and 70% reductions are observed in average inter-story drift values and average acceleration values for RL-Controller, respectively. Next, a comparison is shown between RL-Controller's performance and a well-known optimal control method, LQG, in which the former's superiority in a five story structure is demonstrated. This assessment shows consistent performance advantage in all performance metrics for RL-Controller; e.g., 25% lower inter-story drift ratio, 25% lower peak accelerations and 34% lower story shear is achieved with RL-Controller compared to LQG in one of the discussed actuator layouts. In the last case study, a nonlinear five-story structure with elasto-plastic behavior is numerically simulated to evaluate RL-Controller performance on nonlinear structures. When controlled within all stories, the maximum inter-story drift ratio and acceleration are reduced 72% and 37% respectively. Although, high total acceleration energy values showed recurrently engaged control, which can be eased within training process if preferred by designer. To achieve these significant performance enhancements, a novel synthetic loading scheme is developed and employed during training that mimics real earthquakes. In addition, an adaptive reward function is introduced that stabilizes the constituents of the composite reward function over the course of training. Finally, two essential implementation issues are addressed and evaluated within this framework to show the robustness of the proposed method to (a) system property changes up to ±10%, and (b) control force delays. The results demonstrate the efficacy of the deep reinforcement learning-based paradigm for active structural control. In the future, this research will be expanded to validate the proposed methodology through experimental studies and to consider and address further practical implementation considerations.

CRediT authorship contribution statement

Soheila Sadeghi Eshkevari: Conceptualization, Methodology, Software, Formal analysis, Data curation, Writing, Visualization. **Soheil Sadeghi Eshkevari:** Conceptualization, Methodology, Software, Data curation, Writing, Visualization. **Debarshi Sen:** Conceptualization, Methodology, Data curation, Writing, Visualization. **Shamim N. Pakzad:** Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

Research funding is partially provided by a grant from the Center for Integrated Asset Management for Multimodal Transportation Infrastructure Systems (CIAMTIS), a US Department of Transportation University Transportation Center, and by a grant from the Commonwealth of Pennsylvania, Department of Community and Economic Development, through the Pennsylvania Infrastructure Technology Alliance (PITA) program.

An earlier and limited version of this work has been communicated to IWSHM 2021.

Appendix A. Neural networks architecture design

In this study, we use the Soft Actor–Critic (SAC) algorithm for learning optimal control policies. In general, actor–critic methods require two separate function approximators for value estimation (critic network) and policy determination (actor network). In this study, policy and value functions are modeled as multi-layer perceptrons (MLP) with three hidden layers, each with 128 nodes. This architecture is found by following the ubiquitous approach of trying deep and wide networks initially and then contracting the network size to reach a high accuracy network architecture with fewer number of parameters for reducing total computational cost. The MLP is encompassed between input and output layers that have varying sizes depending on the complexity of the state space and number of designated actions (e.g., number of actuators). In this unified network, the activation functions are chosen as rectified linear unit (ReLU) in each layer except for the last one that has linear activation. Layer normalization is not used in this class. For the optimization, Adam optimizer is used [88]. The temperature parameter α as shown in Eq. (6) to determine the relative importance of entropy term in SAC objective function is a trainable hyperparameter in this framework (initial value 0.1) to be learned optimally throughout the training. The hyperparameters customized for each case study are shown in Table A.7 which includes length of response history in state (l), learning rate (LR), discount factor (γ), episode length, time step (Δt), number of iterations for training, and action range which defines the range (scale) of continuous action space (actuator forces) found between [−1,1]. Note that in inference (testing) using ground motion records, the episode length is adjusted to each earthquake record length.

The computer used for the training is a personal computer with Intel core i7 CPU and ATI V4800 GPU. Training time for the 5-story building is approximately 4 hours.

Appendix B. Ground motion signal generator

In this work, a combination of random noise generation and random impulse generation has been proposed to replicate major earthquake recordings in the training stage for the RL-Controller to learn to react for the unknown forthcoming earthquakes. This load generator is customized according to the mechanical properties of each case study to force the structure up to its designed capacity and beyond that. The random noise generator produces a Gaussian random noise scaled to a given range and the random impulse generator is a *sinc* function shifted by a random number of time intervals. This single random impulse at the random time is added to the random noise for each episode of training as shown in Fig. B.10.

Table A.7
Hyperparameters used to train the RL-Controller for the discussed case studies: length of response history in state (l), learning rate (LR), discount factor (γ), episode length, time interval (Δt), number of iterations, and action range.

Case study	l	LR	γ	Episode len.	$\Delta t(s)$	Num. of itr.	Act. range
3-st frame	5	0.0003	0.999	1000	0.02	3×10^6	5×10^3 lb
5-st building	5	0.00015	0.999	1000	0.01	1.5×10^6	4×10^5 N

References

[1] Spencer Jr B, Nagarajaiah S. State of the art of structural control. J Struct Eng 2003;129(7):845–56.
[2] Zhao D, Lu Z, Zhao H, Li X, Wang B, Liu P. A review of active control approaches in stabilizing combustion systems in aerospace industry. Prog Aerosp Sci 2018;97:35–60.
[3] Block JJ, Strganac TW. Applied active control for a nonlinear aeroelastic structure. J Guid Control Dyn 1998;21(6):838–45.
[4] Elliott SJ. A review of active noise and vibration control in road vehicles. 2008.
[5] Yao G, Yap F, Chen G, Li W, Yeo S. MR damper and its application for semi-active control of vehicle suspension system. Mechatronics 2002;12(7):963–73.
[6] Spencer Jr B, Sain M, Won C-H, Kaspari D, Sain P. Reliability-based measures of structural control robustness. Struct Saf 1994;15(1–2):111–29.
[7] Zhang H, Wang R, Wang J, Shi Y. Robust finite frequency H∞ static-output-feedback control with application to vibration active control of structural systems. Mechatronics 2014;24(4):354–66.
[8] Balas MJ. Direct velocity feedback control of large space structures. J Guid Control 1979;2(3):252–3.
[9] Yang D-H, Shin J-H, Lee H, Kim S-K, Kwak MK. Active vibration control of structure by active mass damper and multi-modal negative acceleration feedback control algorithm. J Sound Vib 2017;392:18–30.
[10] Kober J, Bagnell JA, Peters J. Reinforcement learning in robotics: A survey. Int J Robot Res 2013;32(11):1238–74.
[11] Vecerik M, Hester T, Scholz J, Wang F, Pietquin O, Piot B, Heess N, Rothörl T, Lampe T, Riedmiller M. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. 2017, arXiv preprint arXiv:1707.08817.
[12] Adam B, Smith IF. Reinforcement learning for structural control. J Comput Civ Eng 2008;22(2):133–9.
[13] Housner G, Bergman L, Caughey T, Chassiakos G, Claus R, Masri S, Skelton R, Soong T, Spencer B, Yao T. Structural control: Past, present and future. J Eng Mech ASCE 1997;123(9):897–971.
[14] Buckle I, Mayes R. Seismic isolation: History, application, and performance - A world view. Earthq Spectr 1990;6(2):161–201.
[15] Kelly J. Aseismic base isolation: review and bibliography. Soil Dyn Earthq Eng 1986;5(4):202–16.
[16] Cancellara D, De Angelis F. Assessment and dynamic nonlinear analysis of different base isolation systems for a multi-storey RC building irregular in plan. Comput Struct 2017;180:74–88.
[17] De Luca A, Guidi LG. State of art in the worldwide evolution of base isolation design. Soil Dyn Earthq Eng 2019;125:105722.
[18] Sheikh H, Van Engelen NC, Ruparathna R. A review of base isolation systems with adaptive characteristics. In: Structures, vol. 38. Elsevier; 2022, p. 1542–55.
[19] Zhang R, Soong T. Seismic design of viscoelastic dampers for structural applications. J Struct Eng ASCE 1992;118(5):1375–92.
[20] Feng M, Kim M, Purasinghe R. Viscoelastic dampers at expansion joints for seismic protection of bridges. J Bridge Eng ASCE 2000;5(1):67–74.
[21] Mazza F, Vulcano A. Control of the earthquake and wind dynamic response of steel-framed buildings by using additional braces and/or viscoelastic dampers. Earthq Eng Struct Dyn 2011;40(2):155–74.
[22] Xu ZD, Liao YX, Ge T, Xu C. Experimental and theoretical study of viscoelastic dampers with different matrix rubbers. J Eng Mech 2016;142(8):04016051.
[23] Soto M, Adeli H. Tuned mass dampers. Arch Comput Methods Eng 2013;20:419–31.
[24] Elias S, Matsagar V. Research developments in vibration control of structures using passive tuned mass dampers. Annu Rev Control 2017;44:129–56.
[25] Elias S, Matsagar V. Wind response control of tall buildings with a tuned mass damper. J Build Eng 2018;15:51–60.
[26] Lucchini A, Greco R, Marano G, Monti G. Robust design of tuned mass damper systems for seismic protection of multistory buildings. J Struct Eng 2014;140(8):A4014009.
[27] Lu Z, Wang D, Masri SF, Lu X. An experimental study of vibration control of wind-excited high-rise buildings using particle tuned mass dampers. Smart Struct Syst 2016;18(1):93–115.
[28] Casciati F, Rodellar J, Yildirim U. Active and semi-active control ofstructures – theory and applications: A review of recent advances. J Intell Mater Syst Struct 2012;23(11):1181–95.
[29] Korkmaz S. A review of active structural control: challenges for engineering informatics. Comput Struct 2011;89:2113–32.
[30] Symans M, Constantinou M. Semi-active control systems for seismic protection of structures: a state-of-the-art review. Eng Struct 1999;21(6):469–87.
[31] Dyke S, Spencer B, Sain M, Carlson J. An experimental study of MR dampers for seismic protection. Smart Mater Struct 1998;7(5):693–703.
[32] Jansen L, Dyke S. Semiactive control strategies for MR dampers: Comparative study. J Eng Mech ASCE 2000;126(8):795–803.
[33] Yang G, Spencer B, Carlson J, Sain M. Large-scale MR fluid dampers: modeling and dynamic performance considerations. Eng Struct 2002;24(3):309–23.
[34] Nagarajaiah S, Sahasrabudhe S. Seismic response control of smart sliding isolated buildings using variable stiffness systems: An experimental and numerical study. Earthq Eng Struct Dyn 2006;35:177–97.
[35] Sarlis A, Pasala D, Constantinou M, Reinhorn A, Nagarajaiah S, Taylor D. Negative stiffness device for seismic protection of structures. J Struct Eng ASCE 2013;139(7):1124–33.
[36] Pasala D, Sarlis A, Nagarajaiah S, Reinhorn A, Constantinou M, Taylor D. Adaptive negative stiffness: New structural modification approach for seismic protection. J Struct Eng ASCE 2013;139(7):1112–23.
[37] Nagarajaiah S, Sen D. Apparent-weakening by adaptive passive stiffness shaping along the height of multistory building using negative stiffness devices and dampers for seismic protection. Eng Struct 2020;220:110754.
[38] Nagarajaiah S. Adaptive passive, semi active, smart tuned mass dampers: identification and control using empirical mode decomposition, Hilbert transform, and short-term Fourier transform. Struct Control Health Monit 2009;16(7–8):800–41.
[39] Fujino Y, Sun L, Pacheco B, Chaiseri P. Tuned liquid damper TLD for suppressing horizontal motion of structures. J Eng Mech ASCE 1992;118(10):2017–30.
[40] Yalla S, Kareem A, Kantor J. Semi-active tuned liquid column dampers for vibration control of structures. Eng Struct 2001;23(11):1469–79.
[41] Tait M, Isyumov N, El Damatty A. Performance of tuned liquid dampers. J Eng Mech ASCE 2008;134(5):417–27.
[42] Spencer B, Dyke S, Deoskar H. Benchmark problems in structural control: part II—active tendon system. Earthq Eng Struct Dyn 1998;27(11):1141–7.
[43] Cao H, Reinhorn A, Soong T. Design of an active mass damper for a tall TV tower in Nanjing, China. Eng Struct 1998;20(3):134–43.
[44] Ribakov Y, Gluck J, Reinhorn A. Active viscous damping system for control of MDOF structures. Earthq Eng Struct Dyn 2001;30(2):195–212.
[45] Soong T. State-of-the-art review: Active structural control in civil engineering. Eng Struct 1988;10(2):74–84.
[46] Ikeda Y, Sasaki K, Sakamoto M, Kobori T. Active mass driver system as the first application of active structural control. Earthq Eng Struct Dyn 2001;30(11):1575–95.
[47] Casciati F, Rodellar J, Yildrim U. Active and semi-active control of structures – theory and applications: A review of recent advances. J Intell Mater Syst Struct 2012;23(11):1181–95.
[48] Amini F, Hazaveh N, Rad A. Wavelet PSO-based LQR algorithm for optimal structural control using active tuned mass dampers. Comput-Aided Civ Infrastruct Eng 2013;28:542–57.
[49] Pourzeynali S, Lavasani H, Modarayi A. Active control of high rise building structures using fuzzy logic and genetic algorithms. Eng Struct 2007;29(3):346–57.
[50] Ghaboussi J, Joghataie A. Active control of structures using neural networks. J Eng Mech ASCE 1995;121(4):555–67.
[51] Di Girolamo GD, Smarra F, Gattulli V, Potenza F, Graziosi F, D'Innocenzo A. Data-driven optimal predictive control of seismic induced vibrations in frame structures. Struct Control Health Monit 2020;27(4):e2514. http://dx.doi.org/10.1002/stc.2514, e2514 stc.2514 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/stc.2514 URL https://onlinelibrary.wiley.com/doi/abs/10.1002/stc.2514.
[52] Smarra F, Di Girolamo GD, Gattulli V, Graziosi F, D'Innocenzo A. Learning models for seismic-induced vibrations optimal control in structures via random forests. J Optim Theory Appl 2020;187(3):855–74.
[53] Wiering M, van Otterlo M. Reinforcement learning: State-of-the-art. Springer; 2012.
[54] François-Lavet V, Henderson P, Islam R, Bellemare MG, Pineau J. An introduction to deep reinforcement learning. Found Trends® Mach Learn 2018;11(3–4):219–354. http://dx.doi.org/10.1561/2200000071.
[55] Li Y. Deep reinforcement learning. 2018, CoRR arXiv:1810.06339 URL http://arxiv.org/abs/1810.06339.
[56] Sallab AE, Abdou M, Perot E, Yogamani S. Deep reinforcement learning framework for autonomous driving. Electron Imaging 2017;2017(19):70–6.
[57] Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, Lanctot M, Sifre L, Kumaran D, Graepel T, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. 2017, arXiv preprint arXiv:1712.01815.
[58] Polydoros AS, Nalpantidis L. Survey of model-based reinforcement learning: Applications on robotics. J Intell Robot Syst 2017;86(2):153–73. http://dx.doi.org/10.1007/s10846-017-0468-y.
[59] Nguyen TT, Nguyen ND, Nahavandi S. Deep reinforcement learning for multi-agent systems: A review of challenges, solutions, and applications. IEEE Trans Cybern 2020;50(9):3826–39. http://dx.doi.org/10.1109/TCYB.2020.2977374.

[60] Luong NC, Hoang DT, Gong S, Niyato D, Wang P, Liang Y-C, Kim DI. Applications of deep reinforcement learning in communications and networking: A survey. IEEE Commun Surv Tutor 2019;21(4):3133–74. http://dx.doi.org/10.1109/COMST.2019.2916583.
[61] Mahmud M, Kaiser MS, Hussain A, Vassanelli S. Applications of deep learning and reinforcement learning to biological data. IEEE Trans Neural Netw Learn Syst 2018;29(6):2063–79. http://dx.doi.org/10.1109/TNNLS.2018.2790388.
[62] Eshkevari SS, Eshkevari SS, Pakzad SN, Muñoz-Avila H, Kishore S. Routing of public and electric transportation systems using reinforcement learning. In: Data science in engineering, volume 9. Springer; 2022, p. 263–73.
[63] Khargonekar PP, Dahleh MA. Advancing systems and control research in the era of ML and AI. Annu Rev Control 2018;45:1–4.
[64] Buşoniu L, de Bruin T, Tolić D, Kober J, Palunko I. Reinforcement learning for control: Performance, stability, and deep approximators. Annu Rev Control 2018;46:8–28.
[65] Howell M, Frost G, Gordon T, Wu H. Continuous action reinforcement learning applied to vehicle suspension control. Mechatronics 1997;7(3):263–76.
[66] Adam B, Smith F. Reinforcement learning for structural control. J Comput Civ Eng, ASCE 2006;22(2):133–9.
[67] Kober J, Bagnell J, Peters J. Reinforcement learning in robotics: A survey. Int J Robot Res 2013;32(11):1238–74.
[68] Tang H, Rabault J, Kuhnle A, Wang Y, Wang T. Robust active flow control over a range of Reynolds numbers using an artificial neural network trained through deep reinforcement learning. Phys Fluids 2020;32:053605.
[69] Fan D, Yang L, Wang Z, Triantafyllou M, Karniadakis G. Reinforcement learning for bluff body active flow control in experiments and simulations. Proc Natl Acad Sci 2020;117(42):26091–8.
[70] Haarnoja T, Ha S, Zhou A, Tan J, Tucker G, Levine S. Learning to walk via deep reinforcement learning. 2018, arXiv preprint arXiv:1812.11103.
[71] Kalashnikov D, Irpan A, Pastor P, Ibarz J, Herzog A, Jang E, Quillen D, Holly E, Kalakrishnan M, Vanhoucke V, Levine S. Scalable deep reinforcement learning for vision-based robotic manipulation. In: Billard A, Dragan A, Peters J, Morimoto J, editors. Proceedings of the 2nd conference on robot learning, vol. 87. Proceedings of machine learning research, PMLR; 2018, p. 651–73.
[72] Tai L, Paolo G, Liu M. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS). 2017, p. 31–6. http://dx.doi.org/10.1109/IROS.2017.8202134.
[73] Khalatbarisoltani A, Soleymani M, Khodadadi M. Online control of an active seismic system via reinforcement learning. Struct Control Health Monit 2019;26:e2298.
[74] Rahmani HR, Chase G, Wiering M, Könke C. A framework for brain learning-based control of smart structures. Adv Eng Inf 2019;42:100986.
[75] Sutton RS, Barto AG. Reinforcement learning: An introduction. MIT Press; 2018.
[76] Eshkevari SS, Eshkevari SS, Sen D, Pakzad SN. Structural active control framework using reinforcement learning. Structural Health Monitoring 2021 2021.
[77] Newmark NM. A method of computation for structural dynamics. J Eng Mech Div 1959;85(3):67–94.
[78] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. 2017, arXiv preprint arXiv:1707.06347.
[79] Haarnoja T, Zhou A, Abbeel P, Levine S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: International conference on machine learning. PMLR; 2018, p. 1861–70.
[80] Christodoulou P. Soft actor-critic for discrete action settings. 2019, arXiv preprint arXiv:1910.07207.
[81] Dulac-Arnold G, Mankowitz D, Hester T. Challenges of real-world reinforcement learning. 2019, arXiv preprint arXiv:1904.12901.
[82] PEER NGA database USgs. In: The earthquake engineering online archive. NISEE e-Library.
[83] Chung L, Lin R, Soong T, Reinhorn A. Experimental study of active control for MDOF seismic structures. J Eng Mech 1989;115(8):1609–27.
[84] Park S-K, Noh HY. Updating structural parameters with spatially incomplete measurements using subspace system identification. J Eng Mech 2017;143(7):04017040.
[85] Housner G, Bergman LA, Caughey TK, Chassiakos AG, Claus RO, Masri SF, Skelton RE, Soong T, Spencer B, Yao JT. Structural control: past, present, and future. J Eng Mech 1997;123(9):897–971.
[86] Chung L, Lin C, Lu K. Time-delay control of structures. Earthq Eng Struct Dyn 1995;24(5):687–701.
[87] Chen B, Xu M, Li L, Zhao D. Delay-aware model-based reinforcement learning for continuous control. Neurocomputing 2021;450:119–28.
[88] Kingma DP, Ba J. Adam: A method for stochastic optimization. 2014, arXiv preprint arXiv:1412.6980.