A Framework for Brain Learning-Based Control of Smart Structures

Hamid Radmard Rahmani[a,*], Geoffrey Chase[b], Marco Wiering[c], Carsten Könke[a]

[a] *Institute of Structural Mechanics, Bauhaus-Universität Weimar, Marienstr. 15, D-99423 , Germany*
[b] *Department of Mechanical Engineering, Private Bag 4800, Christchurch 8140, University of Canterbury, New Zealand*
[c] *Institute of Artificial Intelligence and Cognitive Engineering, Faculty of Mathematics and Natural Sciences, University of Groningen, Netherlands*

Abstract

A novel framework for intelligent structural control is proposed using reinforcement learning. In this approach, a deep neural network learns how to improve structural responses using feedback control. The effectiveness of the framework is demonstrated in a case study for a moment frame subjected to earthquake excitations. The performance of the learning method was improved by proposing a state selector function that prevented the neural network from forgetting key states. Results show the controller significantly improved structural responses not only to earthquake records on which it was trained but also to earthquake records new to the controller. The controller also has stable performance under environmental uncertainties. This capability distinguishes the proposed approach and makes it more appropriate for the situations in which it is likely that the controller will be exposed to unpredictable external excitations and high degrees of uncertainties.

Keywords: reinforcement learning, structural control, seismic control, aerospace control, neural networks, deep learning, intelligent control, smart structures, structural dynamics, earthquake

1. Introduction

Seismic control of a structure is a challenging task because of the stochastic nature of earthquakes and their broad frequency content. An advanced solution is the utilization of structural control systems. These systems mitigate vibrations by reducing transmitted forces from the ground to the structure, damping the vibrations, or applying force to the structure in the opposite direction of the earthquake load. Generally, these systems are categorized into passive, active, semi-active, and hybrid systems.

Smart structures can sense their environment and generate the control forces upon that. The main part of the control system in such structures is the control algorithm which determines the behavior of the controller during the external excitations. Some of the recent advances in the control algorithms are summarized by Gutierrez et al. [1].

Classical and optimal control algorithms utilize various methods such as proportional–integral–derivative (PID), linear–quadratic regulator (LQR), linear-quadratic Gaussian (LQG), and fuzzy logic or a combination of them. These systems have been widely utilized in small and large scale systems [2, 3, 4, 5, 6, 7, 8, 9, 10, 11]. Model predictive control (MPC) algorithms use a model to estimate the future evolution of a dynamic process to optimize the control signals to minimize or maximize an objective function [12, 13, 14, 15, 16].

Robust control explicitly deals with uncertainties [17]. In many real control problems, the controller algorithm must deal with a complex system with a high degree of uncertainty. In traditional control

*Corresponding author
Email address: radmard.rahmani@gmail.com (Hamid Radmard Rahmani)

algorithms, the performance of the algorithm is very depended on the accuracy of the considered model of the system which is not available in many real control problems [18, 19, 20].

Intelligent control (IC) is a newer generation of control algorithms utilizing "soft computing" to integrate computational process, reasoning, and decision making along with levels of precisions or uncertainties in the available data, measurements, and the design parameters. Therefore, IC is more realistic for problems with high degrees of uncertainties. The goal of IC is to develop an autonomous system that can operate in an unstructured and uncertain environment independently and without human action [21, 22]. IC uses various artificial intelligence computing approaches like neural networks, Bayesian probability, fuzzy logic [23], machine learning, evolutionary computation, and genetic algorithm [24], as well as the combination of these methods which creates hybrid systems such as neuro-fuzzy [25] or genetic-fuzzy [26]controllers.

Neural controllers are one type of the intelligent controllers [27, 28, 29, 30]. In these studies, Back Propagation(BP) algorithm is typically utilized for offline training. The trained neural network is then tested on new, untrained data. Other approaches such as counter propagation networks (CPN) have also been used [31] but required retraining for new events. Alternatively, *reinforcement learning* (RL), a type of machine learning method, is effectively utilized in different control problems such as traffic signal optimization [32], market-based production control [33], and ship unloader control [34]. It is thus increasingly deployed. Recently, Khalatbarisoltani et al. utilized RL for online tuning an active tuned mass damper (ATMD) which incorporated a fuzzy gain-scheduling controller[35]. Control commands were generated by Proportional Derivative (PD) controller in which gains were tuned by a fuzzy controller, and a Q-table correlates the states to the changes in the fuzzy rule base. The purpose of using the RL algorithm was to improve the fuzzy rule base, using prior experiences.

1.1. Intelligent framework

As a development in the area of intelligent control, this research proposes an intelligent framework that creates an intelligent control system as a trained deep neural network through an automatic process. The method utilized in developing the framework is RL, which has solved some challenging real-world problems [36, 3, 37, 38, 39, 40, 41].

The main advantage of the proposed approach is that it focuses on teaching the controller *How to develop a control policy,* rather than *how to reduce the responses.* This fundamental difference brings more generalization capabilities to the controller. In addition, this approach benefits a model-free method in RL, so the controller doesn't need to know the specific dynamics to develop its control policy which is a significant advantage over model-based methods as the dynamics of the system is not always easy to determine [42].

1.2. Contributions

A novel framework for intelligent control of smart structures is introduced in which a deep neural network would be trained to develop optimum control policy through taking actions and observation cycles. An advanced RL method, called mini-batch learning, is improved based on the characteristics of a structural control problem and the effectiveness of the improved method is demonstrated. The trained controller is shown to result in high levels of generalization and stable performance.

2. System model

2.1. System

In this study, the effectiveness of the developed framework is examined through seismic control of a moment frame as a case study. In the developed approach, the controller doesn't need to know the system's dynamics. Therefore, the developed framework is model-free and could also be utilized for seismic control of multi degree of freedom systems.

The moment frame is modeled as a single degree of freedom (SDOF) system. The mass is 2000 kg, the stiffness is 7.9×10^6 N/m, and the damping is 250×10^3 $N.s/m$. As result, the natural frequency ω, and the period T of the system are 62.84 Hz and 0.1 s respectively.

The Equation of the motion for the system under earthquake excitations and the control forces is:

$$m\ddot{\mathrm{u}} + c\dot{\mathrm{u}} + k\mathrm{u} = -m\ddot{x}_g + f \tag{1}$$

in which m, c, k are the mass, damping, and the stiffness matrices, $\ddot{x}_g$ is the ground acceleration as external excitation, and f is the control force. u, $\dot{u}$, and $\ddot{u}$ are displacement, velocity, and acceleration vectors.

By defining the state vector x:

$$x = \{\mathrm{u},\, \dot{\mathrm{u}}\}^T \tag{2}$$

The state-space representation of the system would be:

$$\dot{x} = Ax + Ff + G\ddot{x}_g \tag{3}$$

$$y_m = C_m x \tag{4}$$

Considering $v = \{\ddot{x}_g,\, f\}$, Equation 3 can be written:

$$\dot{x} = Ax + Bv \tag{5}$$

in which:

$$\begin{aligned} A &= \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -3947.8 & -125.66 \end{bmatrix} \\ B &= \begin{bmatrix} 0 & 0 \\ -1 & \frac{1}{m} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ -1 & 5 \times 10^{-4} \end{bmatrix} \\ C_m &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

3. Reinforcement learning

Reinforcement learning (RL) is an area of machine learning concerned with how software agents ought to take actions in an environment to maximize some notion of cumulative reward [43]. These methods are appropriate for the problems in which the agent can affect the environment. In RL, the environment is typically formulated as a Markov Decision Process (MDP) [44].

An MDP is defined by $\{\mathcal{S},\, \mathcal{A},\, \mathcal{P}^{\mathrm{a}}_{\mathrm{ss'}},\, \mathcal{R}^{\mathrm{a}}_{\mathrm{ss'}},\, \gamma\}$ where $\mathcal{S}$, is the set of states, $\mathcal{A}$ is a set of actions, $\mathcal{P}^{\mathrm{a}}_{\mathrm{ss'}}$is the probability of getting into state s' by taking action a in state s, $\mathcal{R}^{\mathrm{a}}_{\mathrm{ss'}}$ is the corresponding reward, and $\gamma \in [0, 1]$ is a discount factor which adjusts the participation ratio of the future reward in determining the current reward. In the MDPs, the states shall be defined so the current state includes all the required information for decision making about the next actions, which means that "the future is independent of the past given the present".

Based on this definition, a state S_t is Markov if and only if:

$$\mathbb{P}\left[S_{t+1} \mid S_t\right] = \mathbb{P}\left[S_{t+1} \mid S_1, S_2, \ldots, S_t\right] \tag{6}$$

The dynamics in MDPs are represented by *a transition probability* matrix, which correlates state s to a successor, s':

$$P_{ss'} = \mathbb{P}[S_{t+1} = s' \mid S_t = s] \quad (7)$$

In *model-based* reinforcement learning methods, such as dynamic programming, the agent knows the dynamics of the MDP. In the *model-free* methods like Q-learning, the agent doesn't know how the environment works so it learns only by taking action and observing the consequences.

Considering the possibility of taking actions in each state, the policy function, π, represents the probability of taking an action a in state s,

$$\pi(a/s) = \mathbb{P}[A_t\text{=a}|\ S_t\text{=s}] \quad (8)$$

In this paper, an advanced Q-learning method is utilized to train an artificial agent[45]. Using this method, the artificial agent, the intelligent controller, comprising a multi-layer neural network called Q-net, will directly learn by interacting with the environment by applying the control forces and observing the response. From the obtained responses, the learning algorithm assigns the states to the corresponding obtained rewards and correlate them to the action vector including the control forces. The algorithm then improves its control policy to maximize the sum of future rewards. To map a structural control problem into a Q-learning problem, the components of an MDP, including the states, actions, and the reward, are defined in this section.

3.1. States

A proper definition of the state is an important task in RL problems. In order to build an MDP, we have defined the state vector S_t is as follow:

$$\{u_t, u_{t-1}, u_{t-2}, v_t, a_t, \ddot{u}_{g,t}\} \,\epsilon\, S_t \quad (9)$$

where:

u_t : displacement at the time t
v_t : velocity at the time t
a_t : acceleration at the time t
$\ddot{u}_{g,t}$: ground acceleration at the time t

3.2. Actions

In structural control problems, the actions are the control forces. In our case study, the control forces are limited to a realistic and applicable range with an absolute maximum magnitude of $4000N$. The force range is then divided into 40 load-steps to form 40 possible actions in each state. As result, the action-value is a number in the range of $[1, 20]$ if negative, or $[21, 40]$ if positive in direction.

3.3. Reward Function

The reward function plays a critical role in RL problems as it evaluates the behavior of the agent regarding the problem's goals. In this regard, a multi-objective reward function is defined by adding four partial rewards, where targets one objective.

A. Displacement response

The first partial reward function reflects the performance of the controller in terms of reducing the displacement responses:

$$R_{1,t} = 1 - \frac{\mid u_t \mid}{u_{max}}$$

in which, u_t is the displacement value of the frame at the time t and u_{max} is the maximum displacement response.

B. Velocity response

The reward function related to the velocity of the frame is defined as follow:

$$R_{2,t} = 1 - \frac{\mid v_t \mid}{v_{max}}$$

in which v_t is the velocity of the frame at the time $t$and v_{max} is the maximum velocity response.

C. Acceleration response

The controller would be rewarded by reducing the acceleration responses of the frame as follow:

$$R_{3,t} = 1 - \frac{\mid a_t \mid}{a_{max}}$$

in which a_t is the acceleration of the frame at the time t and a_{max} is the maximum acceleration response.

D. Actuator force

The actuator uses electricity to generate force and subsequently producing larger actuation forces requires more power. Therefore, the goal of forth reward function is to reduce the required energy by applying a penalty value equal to 0.005 to the actuator force in each time-step as follow:

$R_{4,t} = f_t \times P_a$

in which:

f_t = Applied actuation force in the time $t(N)$

P_a = Penalty value for unit actuator force (= 0.005)

The overall reward value R, in the time t would be built by combining the four rewards functions:

$$R_t = R_{1,t} + R_{2,t} + R_{3,t} + R_{4,t}$$

3.4. Learning rule

In the Q-learning method, the maximum expected return Q^*, for action a, in the state s, is defined:

$$Q^*(s,a) = \underset{\pi}{max}\, \mathbb{E}\left[r_t + \gamma r_{t+1} + \gamma^2 r_{t+1} + \ldots + \gamma^{n-1} r_n \mid s_t = s, a_t = a, \pi\right] \quad (10)$$

As it is shown, following the policy π, the maximum return Q^* for the given state s and action a, is the sum of reward r, discounted by the factor of γ in each time-step t until the end of the simulation.

An important representation of the optimum action-value function widely utilized in RL problems is called Bellman equation:

$$Q^*(s,a) = \mathbb{E}_{s'}\left[r + \gamma \underset{a}{max} Q^*(s',a') \mid s,a\right] \quad (11)$$

According to the Bellman equation, if the optimum action-values is known for all the actions in the next state s', then the optimum action-value for each action in current state s, is the sum of the immediate reward r and the maximum of the action-value of the next time-step. As a result, the optimum action-values are determined through an iterative process of estimating the optimum values, determining errors and calculating the new values as follow:

$$Q_{i+1}(s,a) = r + \gamma \underset{a}{max} Q(s',a') \quad (12)$$

Using this equation when $i \to \infty$, the Q values will converge Q^*.

In this research, a deep neural network called Q-net estimates the optimum action-values, and generalization also happens during the learning phase as the neural networks are very good in generalization. Having the optimum action-values, the optimum policy π^* takes the action with the maximum action-value in each state.

3.5. Mini-batch learning

In practice, the learning process using Q-net is a challenging task as it can be subjected to instabilities and divergences [46]. Some causes of such instabilities are related to the correlations present in the sequence of observation, updating the Q-values, and taking action so that a small update of Q-values may significantly change the policy, data distribution, and the correlations between the current action-values and the target-values. Volodymyr Mnih et al. [47] addressed these issues by introducing a variant of Q-learning called *Mini-batch learning,* which includes two main improvements:

1. They introduced a biologically inspired experience replay that randomizes over the data and improved the method by breaking the correlations in the training states sequence.
2. They utilized an iterative update method in which the algorithm adjusts the action-values towards target-values periodically but not always. This method breaks the correlation between inputs and outputs of the Q-net during the learning process and prevents related instabilities.

Achieving these goals, they utilized a separate neural network to estimate the optimum action-values, as target-values for training the Q-net, during the learning process. The secondary net is a clone of the Q-net, which updates periodically. They showed the single-agent trained by the developed revision of the Q-learning has better performance.

Initially this research utilized mini-batch learning method for training Q-net, which resulted in controller performance issues. In the next section, such issues are addressed and the method is improved by developing additional functions.

4. Intelligent framework

The body of the developed framework includes three main components shown in Figure 1:

1. Simulink model: Which includes the state-space model of the system and represents the environment.
2. Learning module: Receives the feed-backs from the environment and updates the Q-net.
3. Visualization module: Graphically simulates the system under the external excitation and the control forces and also shows some useful data during training and testing phases.

The Q-net comprises a deep neural network which consists of an input layer with 6 neurons, two hidden layers with 20 neurons, and an output layer with 41 neurons. The *sigmoid* function is the activation function and the net was trained using *backpropagation* methods.

.

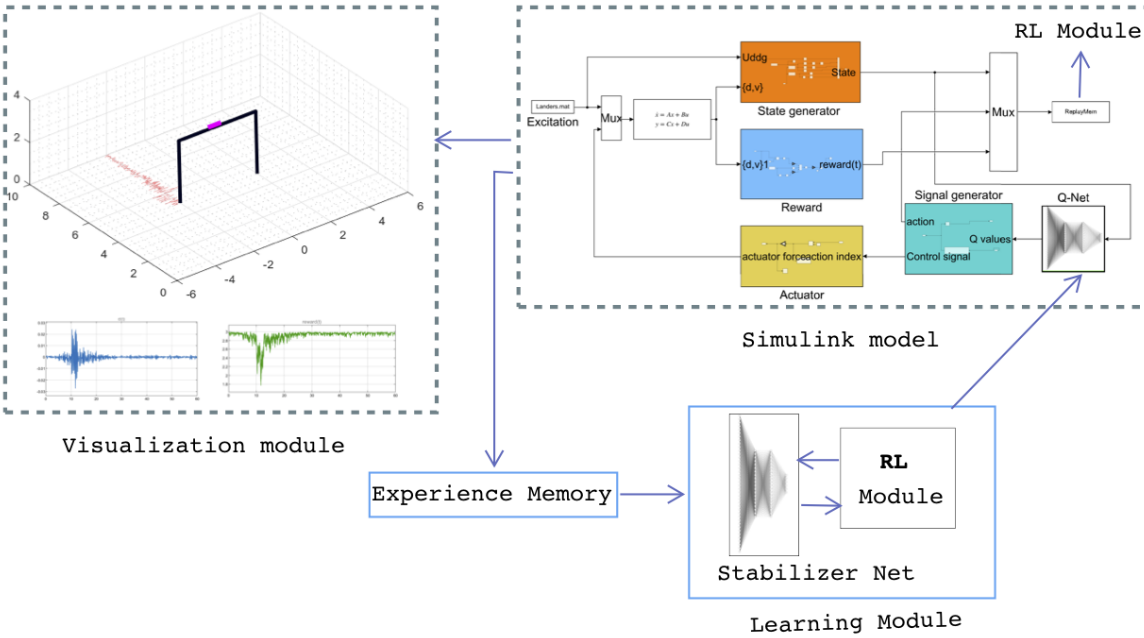
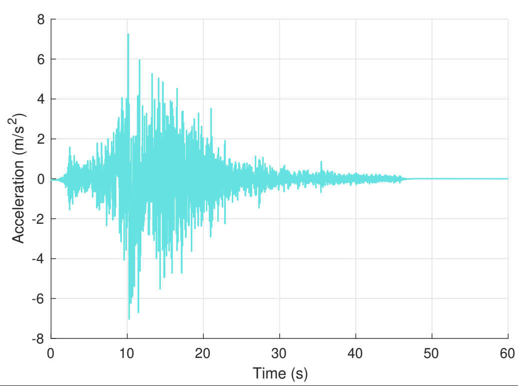


Figure 1: Schematic representation of the intelligent framework

4.1. *Earthquake excitations*

To train the controller, the Landers earthquake record is considered. The acceleration record is obtained from the NGA strong motion database [48]. Figure 2 shows this event and all relevant data.



Max. Acceleration (g)	Max. Velocity (cm/sec)	Max. Displacement (cm)	Effective Design Acceleration (g)	Predominant Period (sec)	Significant Duration (sec)
0.73	141.02	113.78	0.56	0.08	12.92

Figure 2: Landers earthquake record, obtained from Pacific Earthquake Engineering Research (PEER) Center[48]

4.2. *Learning by mini-batch learning method*

Initially, the controller was trained using the mini-batch method[47]. After recording enough states into the experience reply, the target Q-values was determined by the learning module and the Q-net was trained 100 times using the randomly selected *mini-learning batch* of the data. The stabilizer net was then updated every 50 training episodes. The learning parameters are shown in Table 1.

In the beginning, the controller module had no idea about controlling the moment frame under the earthquake excitation and the controlled responses of the frame were worse than uncontrolled. Based on Section 3, the learning algorithm was improved. As it is shown in Figure 3, the average reward is improved from about 2.41 to a maximum of 2.78.

Table 1: Utilized Learning parameters

Number of episodes	Size of experience reply	Number of states per episode	sensor sampling rate (Hz)	Mini-batch size
1000	60000	6000	100	50

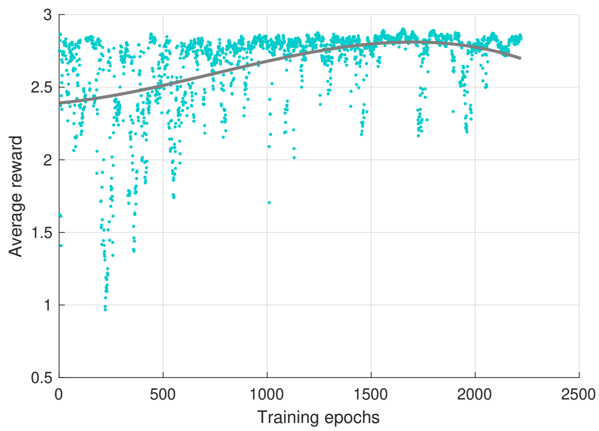


Figure 3: Average reward in learning phase

The algorithm stopped the training after 2200 episodes as no further learning was occurring. The controller was trained to reduce the responses of the frame to the Landers earthquake as shown in Figure 4.

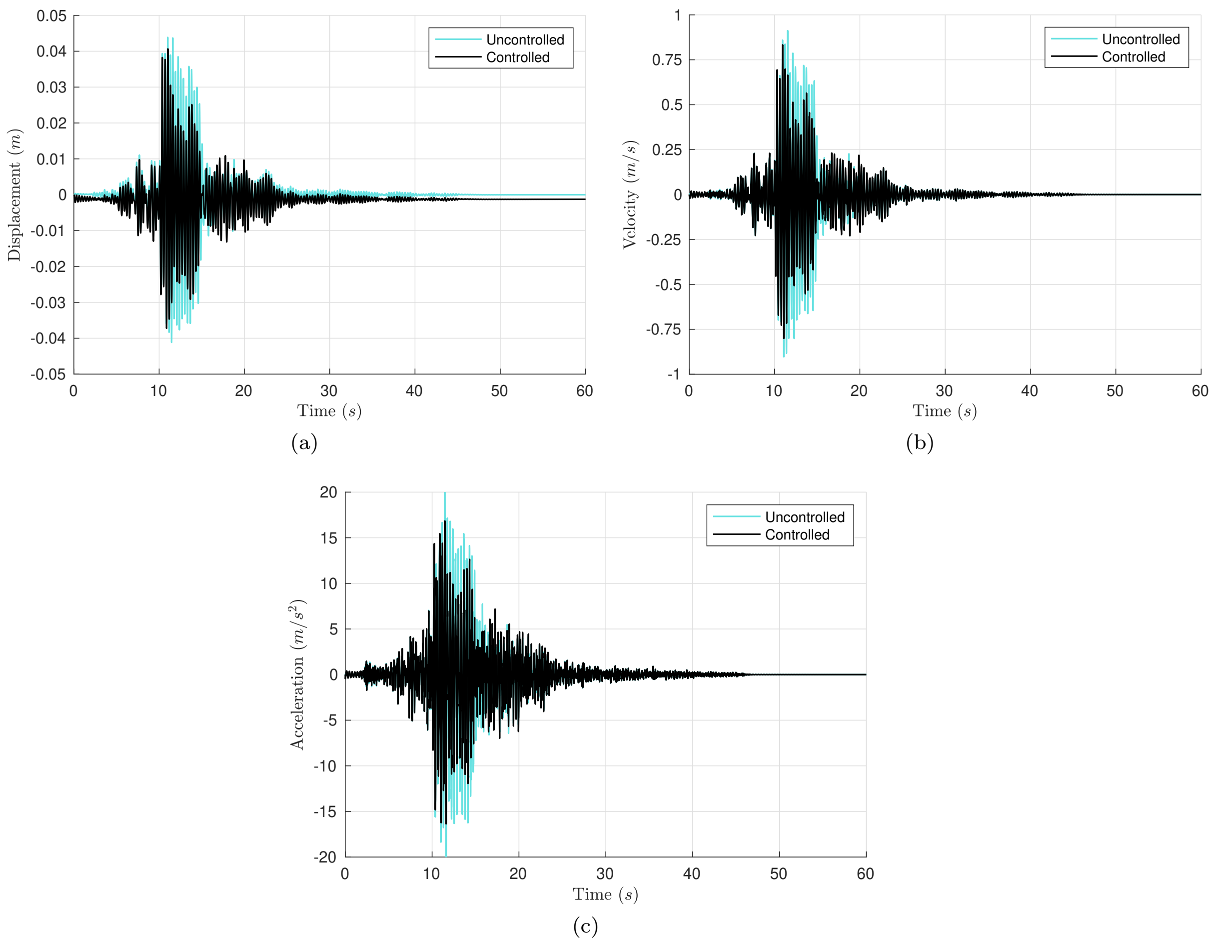


Figure 4: Controlled and uncontrolled responses comparison for Landers earthquake used in training. The controller is trained using the original learning algorithm.

The results show responses are improved by the controller. With respect to reducing the peak responses, the controller showed low performance in reducing peak displacement and velocity responses with the values of 7.1%, 8.7%, respectively, but was better in reducing the peak acceleration responses by 26.7%. Results are in Table 2,

Two main issues were seen indicating the characteristics of the structural control problem were not well considered in the learning method. First, the performance of the controller in reducing the peak responses was not as good as reducing the average responses of the frame. However, reducing the peak responses is an important objective is structural control problems. Second, a residual shifting from the origin was seen in displacement responses, which is not acceptable in practice. It indicates the optimum action-values in the states with very low reward values are not determined properly and, control forces are thus applied to the frame where no significant external excitations exist.

4.3. Improved mini-batch learning

Addressing these issues with the mini-batch learning method, the performance of the method is improved by developing a *batch optimizer* module. This module determines the key states among the experienced states and randomly adds some of them to the training batch, which itself is a collection of the randomly selected states. The key states in this structural control are defined: (1) states with a very low immediate reward value; and (2) states with a low-performance value. The first case includes states which are related to the maximum responses in which the return of the actions are high so it worth trying different actions in such states. The second case includes the states in which differences of the uncontrolled and controlled responses is higher than other states; which occurs in the states where the amplitude of the oscillation is low. The pseudo-code of the module is presented in Algorithm 1.

Algorithm 1 Add key states to the mini learning batch. .

Require: LearningBatch,Controlled Reponses,Uncontrolled Responses
Ensure: LearningBatchKeyStatesAdded

function ADDKEYSTATES($LearningBatch, CR, UCR$)
  $SortedBatch \leftarrow SORT(LearningBatch, Rewards)$
  $TargetSt1 \leftarrow RANDOMSELECT(SortedBatch, 10, 100)$ ▷ randomly selects 10 state among the first 100 states with minimum reward
  $Performance \leftarrow UCR - CR$
  $CurrentLearningBatch \leftarrow ADD(LearningBatch, Performance)$
  $SortedBatch \leftarrow SORT(LearningBatch, Performance)$
  $TargetSt2 \leftarrow RANDOMSELECT(SortedBatch, 10, 100)$ ▷ randomly selects 10 state among the first 100 states with minimum performance
  $BatchAddedKeyStates \leftarrow ADD(LearningBatch, TargetSt1)$
  $BatchAddedKeyStates \leftarrow ADD(LearningBatch, TargetSt2)$
return $BatchAddedKeyStates$
end function

The improved method is then examined to train the net. As it is shown in Figure 5, the stop-function allowed the training episodes to reach 11000 because of the incremental progress of the performance. The average reward is increased from about 2.61 to a value of 2.90 which indicates better performance of the learning algorithm.

To show the effectiveness of the improved learning algorithm, the uncontrolled and controller responses under both methods are compared in Figure 6. The results indicate the improved method upgraded the performance of the controller in reducing peak responses, as well as average responses. In addition, the issue about the shifting from the origin, which was seen in initial results, is solved.

As summarized in Table 2, the improved algorithm significantly upgraded the performance of the controller in reducing the peak displacement response from 7.1% to 46%. For t velocity, the performance improved from to 8.7% to 41%, and for acceleration, performance improved from 26.7% to 37.8%. Similar upgrades in performance of the method are also seen in terms of reducing average RMS responses.

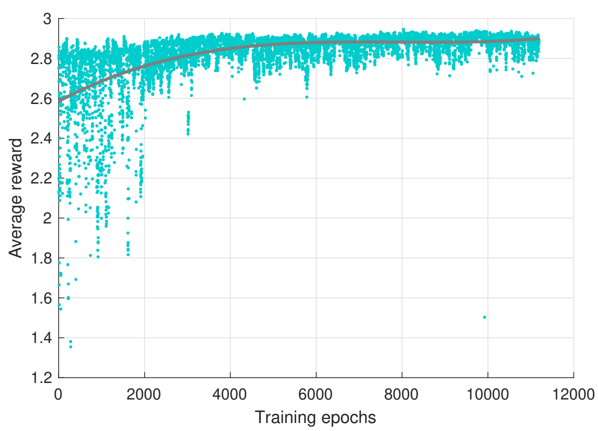
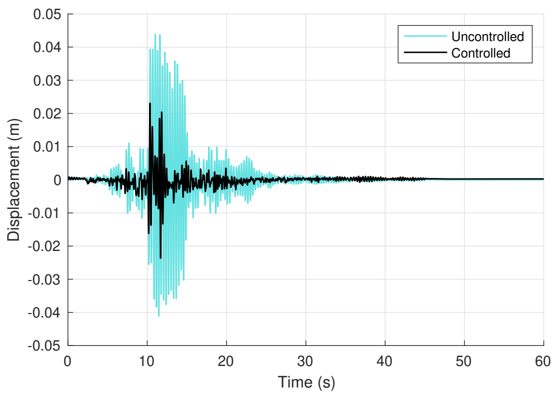
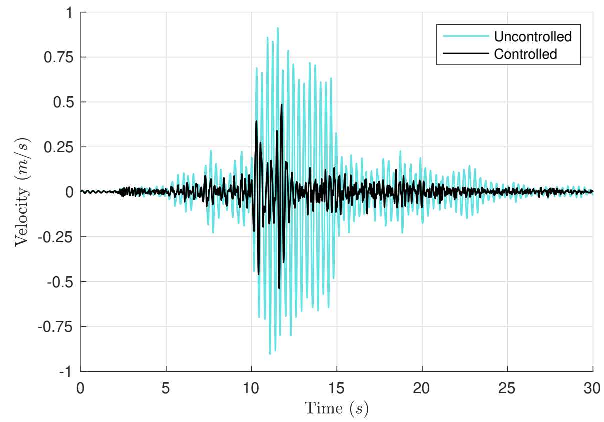


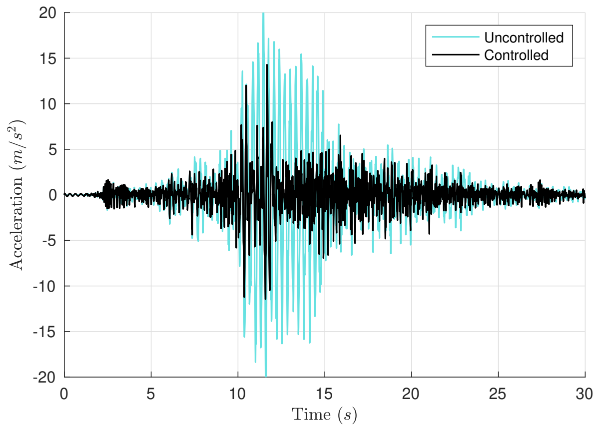
Figure 5: Improvement of the average rewards during the learning process using the improved method.



(a) Landers - displacement



(b) Landers - velocity



(c) Landers - acceleration

Figure 6: Comparison of the controlled and uncontrolled responses of the frame when the controller was learned by the improved method.

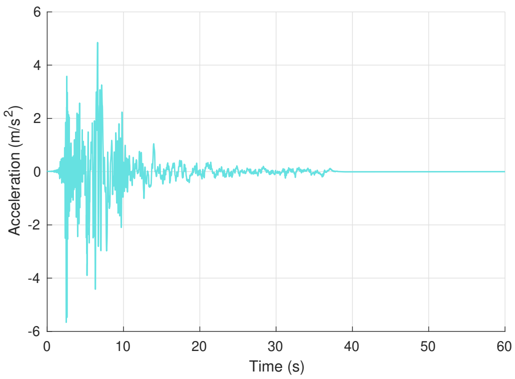
Learning method		Uncontrolled	Controlled	Improvement
Original method	Peak Dis.	4.39	4.06	7.1%
	Peak Vel.	0.91	0.83	8.7%
	Peak Acc.	22.97	16.84	26.7 %
	RMS Dis.	0.0081	0.0068	15.5 %
	RMS Vel.	0.118	0.097	17.7 %
	RMS Acc.	3.378	2.816	16.6 %
Improved method	Peak Dis.	4.39	2.36	46.1 %
	Peak Vel.	0.91	0.54	41.0 %
	Peak Acc.	22.97	14.28	37.8 %
	RMS Dis.	0.0081	0.0017	78.8%
	RMS Vel.	0.118	0.0319	72.9%
	RMS Acc.	3.378	1.180	65.0

Table 2: Comparing the controlled and uncontrolled responses to Landers earthquake excitations when the controller was learnt using original and improved methods (Dis. = Displacement (cm), Vel. = Velocity (m/s), Acc. = Acceleration (m/s^2)).

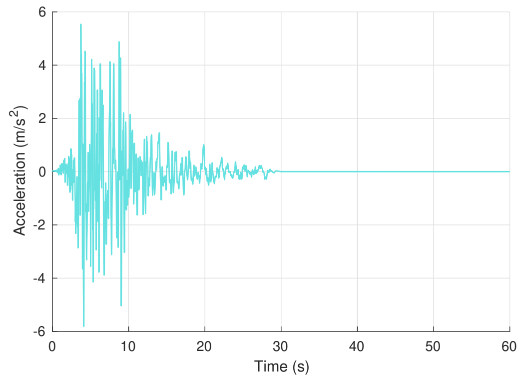
4.4. Testing the controller

To study the performance of the intelligent controller, the trained controller was tested under four new scaled earthquake records obtained from the NGA strong motion database [48](see Figure 7). The controlled and uncontrolled responses are compared in the Figure 8. The controller has effectively improved the frame responses in all cases. As is summarized in the Table 3, in reducing the displacement responses, the maximum performance is 52.5% under the Northridge earthquake and the minimum observed performance is a 40.9% reduction for the Kobe earthquake. The performance of the controller in reducing velocity responses varies between 41.4% to 56.3%. Acceleration responses are improved between 37.3% to 50.1%.

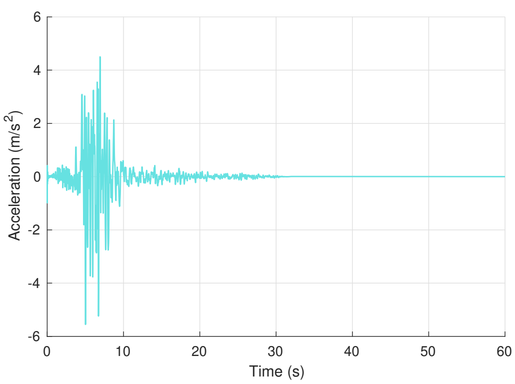
The average performance of the controller in terms of improving displacement responses under four new earthquakes is 47.1% which is comparable to the 46 % obtained for the Landers earthquake, the earthquake for which it was trained. The average performance for reducing velocity and accelerations are 49.2% and 43.4%, respectively, which are even higher than obtained for the Landers earthquake, which are 41.0 % and 37.8 %. In addition, the average RMS of displacement, velocity, and acceleration responses are significantly improved by 57.6 %, 67.9 %, and 46.6 %, respectively (Table 3).



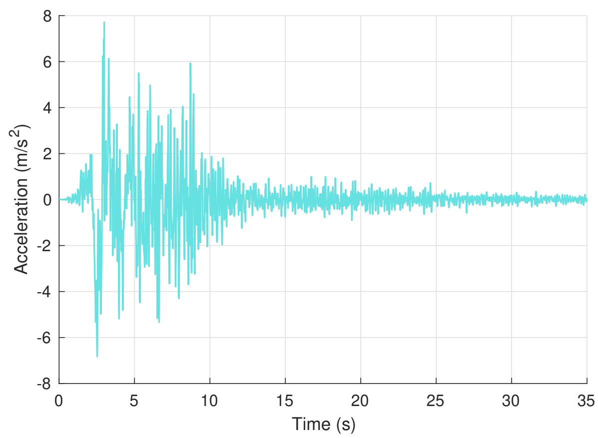
(a) El Centro, 1940- USA



(b) Northridge, 1971- USA



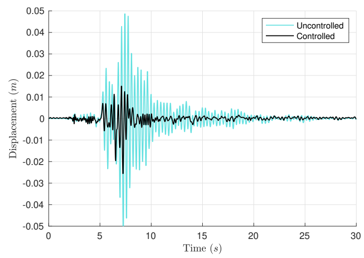
(c) Kobe, 1995- Japan



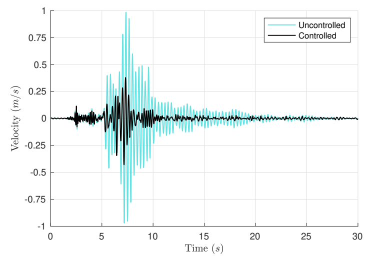
(d) Bam, 2003- Iran

Accelerogram	Max. Acceleration (g)	Max. Velocity (cm/sec)	Max. Displacement (cm)	Effective Design Acceleration (g)	Predominant Period (sec)	Significant Duration (sec)
1- Bam	0.78	128.55	33.95	0.64	0.20	8.24
2- El Centro	0.57	75.76	28.36	0.48	0.32	9.16
3- Kobe	0.56	38.83	15.68	0.56	0.36	4.16
4- Northridge	0.59	64.57	22.16	0.58	0.40	10.40

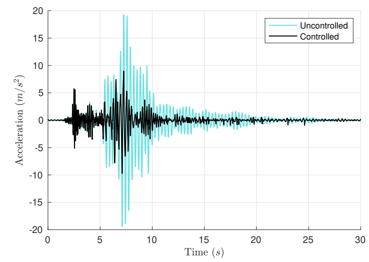
Figure 7: Four earthquake acceleration records which were considered to test the performance of the controller



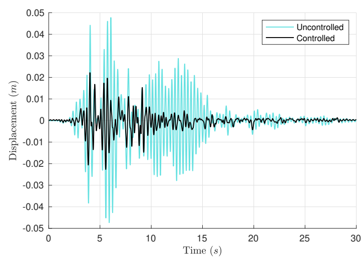
(a) El Centro - displacement



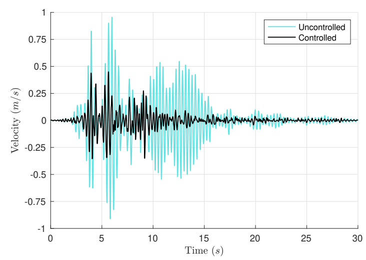
(b) El Centro - velocity



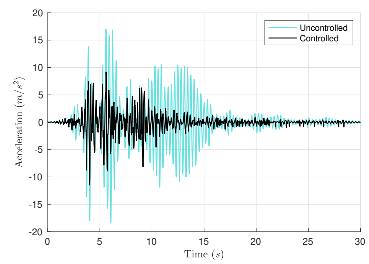
(c) El Centro - acceleration



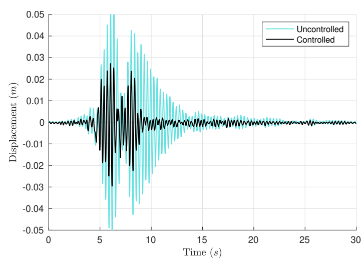
(d) Northridge - displacement



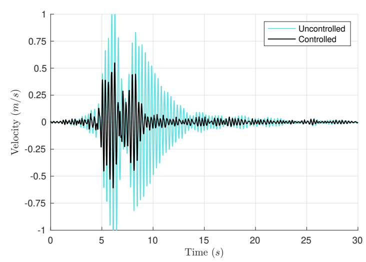
(e) Northridge - velocity



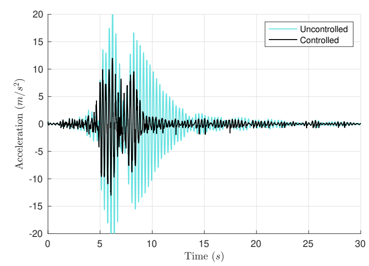
(f) Northridge - acceleration



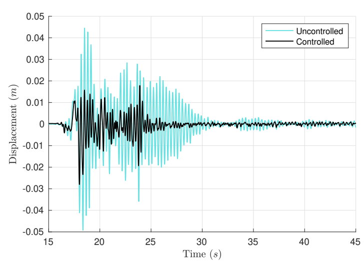
(g) Kobe - displacement



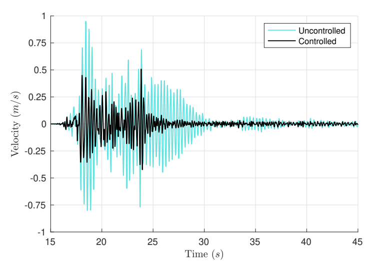
(h) Kobe - velocity



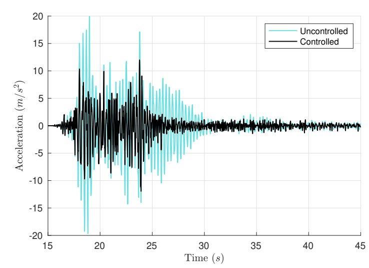
(i) Kobe - acceleration



(j) Bam - displacement



(k) Bam - velocity



(l) Bam - acceleration

Figure 8: Uncontrolled and controlled responses of the frame to the test earthquake excitations which are new to the intelligent controller .

Earthquakes		Uncontrolled	Controlled	Improvement
El Centro	Dis.	5.35	2.56	52.1 %
	Vel.	0.98	0.43	56.3 %
	Acc.	19.46	9.71	50.1 %
	RMS Dis.	0.0045	0.0010	77.9 %
	RMS Vel.	0.1178	0.0368	713 %
	RMS Acc.	2.0802	0.7190	65.4 %
Northridge	Dis.	4.77	2.27	52.5 %
	Vel.	0.95	0.45	52.9 %
	Acc.	18.28	11.46	37.3 %
	RMS Dis.	0.0067	0.0030	55.6 %
	RMS Vel.	0.1721	0.0399	76.8 %
	RMS Acc.	2.3172	1.4011	39.5 %
Kobe	Dis.	5.91	3.49	40.9 %
	Vel.	1.15	0.68	41.4 %
	Acc.	24.24	13.30	45.1 %
	RMS Dis.	0.0080	0.0049	38.3 %
	RMS Vel.	0.1498	0.0560	62.6 %
	RMS Acc.	2.8114	2.0754	26.1 %
Bam	Dis.	4.92	2.80	43.1 %
	Vel.	0.95	0.50	46.5 %
	Acc.	20.44	11.99	41.3 %
	RMS Dis.	0.0083	0.0044	46.4 %
	RMS Ve.	0.1304	0.0559	57.1 %
	RMS Acc.	3.1137	1.9390	37.7 %
Average response	Dis.	5.24	2.78	47.1 %
	Vel.	1.01	0.52	49.2 %
	Acc.	20.61	11.62	43.4 %
	RMS Dis.	0.0030	0.0071	57.6 %
	RMS Vels.	0.0441	0.1376	67.9 %
	RMS Acc.	1.4629	2.7403	46.6 %

Table 3: Responses to the earthquake excitations (Dis. = Displacement (cm), Vel. = Velocity (m/s), Acc. = Acceleration (m/s^2)).

5. Environmental uncertainties

Stable performance under uncertainties is an important advantage. To show the effectiveness of the controller under such uncertainties, the stiffness matrix was multiplied to an uncertainty factor which varies between 5 % and 40 %, and the resultant system is then subjected to the El Centro earthquake. Note that the active control systems are not sensitive to the uncertainties in the damping matrix [49]. As presented in Table 4, the obtained performance varies between 37.2% to 60.3% with an average value of 51.8%, which is very close to the value of 52.1%, obtained without considering uncertainties under El Centro earthquake.

Uncertainty	Uncontrolled Dis.	Controlled Dis.	Improvement (%)
-5 %	0.0535	0.02559	52.1
+5 %	0.0537	0.02291	57.3
-10 %	0.0521	0.02699	48.2
+10 %	0.0493	0.01970	60.0
-15 %	0.0518	0.03016	41.8
+15 %	0.0411	0.01932	53.1
-20 %	0.0558	0.03507	37.2
+20 %	0.0401	0.01778	55.6
-30 %	0.0649	0.03682	43.2
+30 %	0.0429	0.01766	58.8
-40 %	0.0728	0.03350	54.0 f
	Average		51.8

Table 4: Performance of the controller in reducing displacement responses under environmental uncertainties (Dis. = Displacement (m))

6. Conclusion

This paper studied a novel brain learning-based and model-free control approach for smart structures. In this approach, a deep neural network learns how to mitigate the vibrations of a dynamic system subjected to the earthquake excitations. The issues with the current RL method in such problem are addressed and the method is improved. As a case study, the framework was examined for seismic control of a moment frame, subjected to four new earthquake excitations. Moreover, the performance of the controller in the existence of environmental uncertainties was studied. Considering the obtained results and discussions in the previous sections, the following conclusions have been drawn:

1. The developed framework is able to train a deep neural network to significantly improve the responses of a dynamic system to the earthquake excitations.
2. Considering the obtained results during train and test phases, it is concluded that the controller is capable to retain its performance under the earthquake excitations for which it is not trained.
3. The controller showed a stable performance under environmental uncertainties which implies the applicability of the proposed approach in real situations.
4. Learning the controller by the original mini-batch learning method resulted in poor performance in reducing the peak displacement and velocity responses. In addition, shifting from origin was observed in the displacement responses of the frame.
5. The proposed improved mini-batch learning method solves the addressed issues with the original method and significantly improved the performance of the controller. The improved method is applicable to similar problems which suffer such issues.

Declarations of interest

None

References

[1] Mariantonieta Gutierrez Soto and Hojjat Adeli. Recent advances in control algorithms for smart structures and machines. *Expert Systems*, 34(2):e12205, 2017.

[2] Richard C. Dorf and Robert H. Bishop. *Modern Control Systems*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 9th edition, 2000.

[3] Richard C. Dorf and Andrew Kusiak. Handbook of Design, Manufacturing and Automation. https://www.wiley.com/en-us/Handbook+of+Design%2C+Manufacturing+and+Automation-p-9780471552185.

[4] Mark J. Balas. Direct velocity feedback control of large space structures. *Journal of Guidance, Control, and Dynamics*, 2(3):252–253, 1979.

[5] Steven Ian Moore and SO Reza Moheimani. Vibration control with MEMS electrostatic drives: A self-sensing approach. *IEEE Transactions on Control Systems Technology*, 23(3):1237–1244, 2015.

[6] Yoshiki Ikeda, Katsuyasu Sasaki, Mitsuo Sakamoto, and Takuji Kobori. Active mass driver system as the first application of active structural control. *Earthquake Engineering & Structural Dynamics*, 30(11):1575–1595.

[7] Huibert Kwakernaak and Raphael Sivan. *Linear Optimal Control Systems*, volume 1. Wiley-interscience New York, 1972.

[8] S. N. Deshmukh and N. K. Chandiramani. LQR Control of Wind Excited Benchmark Building Using Variable Stiffness Tuned Mass Damper. *Shock and Vibration*, 2014:1–12, 2014.

[9] Luciana R. Barroso, J. Geoffrey Chase, and Stephen Hunt. Resettable smart dampers for multi-level seismic hazard mitigation of steel moment frames. *Journal of Structural Control*, 10(1):41–58, 2003.

[10] J. Geoffrey Chase, Luciana R. Barroso, and Stephen Hunt. The impact of total acceleration control for semi-active earthquake hazard mitigation. *Engineering Structures*, 26(2):201–209, 2004.

[11] J. Geoffrey Chase, Luciana R. Barroso, and Stephen Hunt. Quadratic jerk regulation and the seismic control of civil structures. *Earthquake engineering & structural dynamics*, 32(13):2047–2062, 2003.

[12] Josue Enríquez-Zárate1 Guillermo Valencia-Palomo, Francisco-Ronay López-Estrada, Gerardo Silva-Navarro, and José Antonio Hoyo-Montaño. Efficient predictive vibration control of a building-like structure. 2019.

[13] Ahsan Kareem, Mei Gang, and Jeffrey Kantor. Model predictive control of wind-excited building: Benchmark study. *Journal of Engineering Mechanics*, 130(4):459–465, 2004.

[14] Arne Koerber and Rudibert King. Combined feedback–feedforward control of wind turbines using state-constrained model predictive control. *IEEE Transactions on Control Systems Technology*, 21(4):1117–1128, 2013.

[15] S. Joe Qin and Thomas A. Badgwell. A survey of industrial model predictive control technology. *Control engineering practice*, 11(7):733–764, 2003.

[16] Stefano Riverso, Marcello Farina, and Giancarlo Ferrari-Trecate. Plug-and-play decentralized model predictive control for linear systems. *IEEE Transactions on Automatic Control*, 58(10):2608–2614, 2013.

[17] William S. Levine. *Control System Fundamentals*. CRC press, 2019.

[18] Zhijun Li and Hojjat Adeli. New discrete-time robust H2/H∞ algorithm for vibration control of smart structures using linear matrix inequalities. *Engineering Applications of Artificial Intelligence*, 55:47–57, 2016.

[19] J. Geoffrey Chase and H. Allison Smith. Robust H∞ control considering actuator saturation. I: Theory. *Journal of Engineering Mechanics*, 122(10):976–983, 1996.

[20] J. Geoffrey Chase, H. Allison Smith, and Tetsuo Suzuki. Robust H∞ control considering actuator saturation. II: Applications. *Journal of Engineering Mechanics*, 122(10):984–993, 1996.

[21] Alexander S. Poznyak, Edgar N. Sanchez, and Wen Yu. *Differential Neural Networks for Robust Nonlinear Control: Identification, State Estimation and Trajectory Tracking*. World Scientific, 2001.

[22] Grant H. Kruger, Albert J. Shih, Danie G. Hattingh, and Theo I. van Niekerk. Intelligent machine agent architecture for adaptive control optimization of manufacturing processes. *Advanced Engineering Informatics*, 25(4):783–796, 2011.

[23] Clarence W. De Silva. *Intelligent Control: Fuzzy Logic Applications*. CRC press, 2018.

[24] Ahmad Taher Azar and Sundarapandian Vaidyanathan. *Computational Intelligence Applications in Modeling and Control*. Springer, 2015.

[25] Jyh-Shing Roger Jang, Chuen-Tsai Sun, and Eiji Mizutani. Neuro-fuzzy and soft computing-a computational approach to learning and machine intelligence [Book Review]. *IEEE Transactions on automatic control*, 42(10):1482–1484, 1997.

[26] Mo Jamshidi and Ali Zilouchian. *Intelligent Control Systems Using Soft Computing Methodologies*. CRC press, 2001.

[27] Bani-Hani Khaldoon and Ghaboussi Jamshid. Nonlinear Structural Control Using Neural Networks. *Journal of Engineering Mechanics*, 124(3):319–327, March 1998.

[28] HM Chen, KH Tsai, GZ Qi, JCS Yang, and F Amini. Neural network for structure control. *Journal of Computing in Civil Engineering*, 9(2):168–176, 1995.

[29] T. K. Chang, Ding-Li Yu, and Ding-Wen Yu. Neural network model adaptation and its application to process control. *Advanced Engineering Informatics*, 18(1):1–8, 2004.

[30] Jamshid Ghaboussi and Abdolreza Joghataie. Active control of structures using neural networks. *Journal of Engineering Mechanics*, 1995.

[31] Alok Madan. Vibration control of building structures using self-organizing and self-learning neural networks. *Journal of sound and vibration*, 287(4):759–784, 2005.

[32] Mohammad Aslani, Stefan Seipel, Mohammad Saadi Mesgari, and Marco Wiering. Traffic signal optimization through discrete and continuous reinforcement learning with robustness analysis in downtown Tehran. *Advanced Engineering Informatics*, 38:639–655, 2018.

[33] Balázs Csanád Csáji, László Monostori, and Botond Kádár. Reinforcement learning in a distributed market-based production control system. *Advanced Engineering Informatics*, 20(3):279–288, 2006.

[34] Leonardo Azevedo Scardua, José Jaime Da Cruz, and Anna Helena Reali Costa. Optimal control of ship unloaders using reinforcement learning. *Advanced engineering informatics*, 16(3):217–227, 2002.

[35] Arash Khalatbarisoltani, Mehdi Soleymani, and Masoud Khodadadi. Online control of an active seismic system via reinforcement learning. *Structural Control and Health Monitoring*, 0(0):e2298.

[36] Andrew Y. Ng, Adam Coates, Mark Diel, Varun Ganapathi, Jamie Schulte, Ben Tse, Eric Berger, and Eric Liang. Autonomous Inverted Helicopter Flight via Reinforcement Learning. In Marcelo H. Ang and Oussama Khatib, editors, *Experimental Robotics IX*, Springer Tracts in Advanced Robotics, pages 363–372. Springer Berlin Heidelberg, 2006.

[37] Marco Wiering and Martijn van Otterlo. *Reinforcement Learning: State-of-the-Art*. Springer Science & Business Media, March 2012.

[38] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *arXiv:1712.01815 [cs]*, December 2017.

[39] Patrick Mannion, Jim Duggan, and Enda Howley. Parallel Reinforcement Learning for Traffic Signal Control. *Procedia Computer Science*, 52:956–961, 2015.

[40] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, page 0278364913495721, 2013.

[41] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, October 2017.

[42] C. Zhou, G. W. Rodgers, J. B. Mander, and J. G. Chase. Multi-Event, Multi-Structure Experimental Validation of Hysteresis Loop Analysis SHM. 2018.

[43] Richard S. Sutton and Andrew G. Barto. The MIT Press, 2012.

[44] Andrew G. Barto Richard S. Sutton. *Reinforcement Learning, an Introduction*. The MIT Press, 1998.

[45] C.J. Watkins and P. Dayan. Q-learning. *Mach.Learn.*, (8):279–292, 1992.

[46] J. N. Tsitsiklis and B. Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, May 1997.

[47] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015.

[48] PEER Ground Motion Database - PEER Center. https://ngawest2.berkeley.edu/.

[49] J. N. Yang and A. Akbarpour. Effect of system uncertainty on control of seismic-excited buildings. *Journal of Engineering Mechanics*, 116(2):462–478, 1990.