Journal Européen des Systèmes Automatisés
Vol. 57, No. 2, April, 2024, pp. 433-441
Journal homepage: http://iieta.org/journals/jesa

Intelligent Control of Building Vibrations: A Transformer-Based Deep Reinforcement Learning Framework

Imad Z. Gheni*, Hussein M.H. Al-Khafaji, Hassan M. Alwan

Mechanical Engineering Department, University of Technology-Iraq, Baghdad 10066, Iraq

Corresponding Author Email: imad.zuhair@jmu.edu.iq

https://doi.org/10.18280/jesa.570213

Received: 8 January 2024
Revised: 30 March 2024
Accepted: 15 April 2024
Available online: 28 April 2024

Keywords:
vibration control, deep reinforcement learning (DRL), LQR model, state space, proximal policy optimization, transformer method

ABSTRACT

Deep reinforcement learning (DRL) has emerged as a promising methodology for optimizing control policies across diverse domains, despite its well-acknowledged high training costs. This paper delves into the application of transformer model-based DRL for vibration control in building structures. Specifically, we tackle the challenge of diminishing vibrations induced by external factors like wind or earthquakes. Our innovative method eliminates the necessity for online interaction with the simulation environment during training, offering a more resource-efficient approach. In our proposed framework, the DRL agent learns to dynamically adjust the control signal of a classical linear–quadratic regulator (LQR)-based model in real-time to alleviate building structure vibrations. Combining the proximal policy optimization (PPO) method with a deep neural network trained on experimental environment data using the transformer model, our approach utilizes input sensor data obtained from the structure. The DRL model then generates corrective signals that augment the LQR model's output. Through an experimental study on a small-scale 3-story building structure, we demonstrate the efficacy of our transformer-based DRL control. Our results highlight the superiority of our approach over the classical LQR model in terms of both training computational cost and vibration reduction. This underscores the potential of DRL in enhancing the functionality of construction frameworks when facing external disturbances. Moreover, our adaptable framework is simple to include in the building control systems now in use. with the potential for extension to various control challenges within the realm of structural engineering.

1. INTRODUCTION

Buildings, intricate systems exposed to a myriad of external forces such as wind, seismic activities, and human interactions, undergo dynamic vibrations that can significantly impact occupant comfort and potentially compromise the structural integrity of the edifice. The imperative to control these vibrations is paramount for ensuring the safety, comfort, and longevity of structures. Over the years, researchers and engineers have dedicated extensive efforts to investigate and develop effective control methodologies, leading to the categorization of three primary control methods: passive, semi-active, and active [1].

Passive control methods, leveraging devices like tuned mass dampers, viscoelastic materials, and base isolation systems, present cost-effective solutions for the dissipation of energy and reduction of vibration amplitudes [2]. These methods are particularly suitable for retrofitting existing structures, given their relative affordability and ease of installation. On the other end of the spectrum, the methods of active control involve systems of feedback control equipped with actuators, such as piezoelectric materials, hydraulic actuators, or electromagnetic shakers. These systems dynamically adjust the control signal to counteract vibrations in real time, resulting in high levels of vibration reduction [3-5]. Semi-active control methods strike a middle ground, utilizing smart materials like magnetorheological (MR) fluids to adjust damping force or stiffness in real-time based on measured vibration levels. This approach offers a balanced compromise between the efficiency of systems in action and the ease of use of passive systems [6-8].

While passive and semi-active systems are acknowledged for their affordability and dependability, they do have limitations, particularly in their ability to adapt to diverse types of excitations. Active mass dampers, active tendon systems, and active brace systems are examples of active control systems. excel at obtaining high-performance outcomes by utilising real-time observations to calculate the actuator control force that is required [3, 5, 9]. have been used to achieve precise control forces utilising sensor readings. These techniques include sliding mode control (SMC), pole assignment, and LQR [10-13].

Recent advancements in machine learning have revolutionized various industries, including structural engineering. Machine learning algorithms, particularly those based on deep learning (DL), have proven highly effective in addressing complex challenges. By leveraging data collected from building sensors, these algorithms can analyze large volumes of information. One specific application is the identification of optimal control strategies to mitigate vibrations, a technique known as data-driven control. Data-driven control has the potential to significantly enhance the efficiency and effectiveness of traditional control methods. The collected data encompasses a wide range of information, such as the building's structural properties, external forces acting upon it, and the vibrations themselves. Machine learning algorithms process this data to construct predictive models that optimize the control system and minimize vibrations. These models take into account factors like the building's structural properties, external forces, and the effectiveness of different control strategies. One crucial advantage of this approach is the ability to update the models in real-time as new data becomes available. This dynamic adaptation empowers the control system to respond and adjust to changing conditions promptly. Ultimately, the development of machine learning, particularly deep learning algorithms, has opened up new avenues for effectively addressing complex issues in structural engineering and other sectors. DRL, a subset of machine learning, has recently exhibited promising results in the domain of structural vibration control. Notably, DRL leverages neural networks with multiple layers, known as DL, to enhance its capabilities [14]. For instance, Rahmani et al. [15] A cutting-edge approach known as the deep Q-network (DQN) has been proposed to enhance structural responses through feedback control. This method has shown remarkable effectiveness in reducing structural vibrations caused by earthquake excitations. In a study conducted by Kim and Kim [16], a deep DQN was employed to implement semi-active control on a representative building structure. The results demonstrated a significant reduction in seismic response. Building upon this research, Zhang and Zhu [17] developed a deep deterministic policy gradient-based model to regulate vibrations in single and multi-degree-of-freedom shear-building models. Their findings revealed that this model achieved comparable outcomes to the classical linear quadratic regulator (LQR) approach and exhibited superior performance in a partially observed system. These advancements highlight the potential of deep learning techniques, such as DQN and deep deterministic policy gradients, in improving control strategies for structural dynamics. While these studies showcase commendable results, they highlighted the main limitation of using DRL for control problems, which is the training time that is representative of computational cost. Here, DRL models need to interact with the simulation environments for construable time to achieve an accurate control policy. In this context, the present paper introduces a DRL approach to control building vibration without the need for online interaction with the simulation environment during the training. This goal is achieved in this study by coupling the DRL algorithm with a transformer model, which is a supervised DL model that can map sequential data. While the transformer model is intended to enhance the performance of a traditional LQR control algorithm, the DRL model applies the DRL technique. This integration is driven not only by a large reduction in the computational cost associated with DRL techniques but also by a practical improvement in seismic response. This paper aims to address the specific research gaps compared to previous work During training time and computational cost previous, studies have highlighted the limitation of using DRL for control problems, which is the extensive training time and computational cost associated with interacting with simulation environments. The present paper proposes a DRL approach that does not require online interaction with the simulation environment during training. This addresses the need for more efficient training methods to reduce computational requirements by applying the Integration of DRL with a Transformer Model, where the coupling of DRL algorithm with a transformer model, which is a supervised DL model capable of mapping sequential data. This integration aims to enhance the performance of a classical LQR control algorithm by utilizing the capabilities of both DRL and transformer models. This addresses the gap in research regarding the combined use of DRL and transformer models for structural vibration control.

The remainder of this paper is organized as follows. Section 2 explains the methodology of this study. Section 3 explains the experimental setup and the simulation of building vibrations. Section 4 discusses the results of testing the proposed approach. Finally, Section 5 presents the conclusions of this study.

2. METHODOLOGY

2.1 DRL

Reinforcement learning (RL) distinguishes itself from supervised and unsupervised learning by utilizing the Markov decision process, which involves a cyclic interaction between an agent and its environment. This iterative process consists of four fundamental components: state (*s*), action (*a*), policy (π(*a*|*s*)), and reward (*r*). In the realm of DRL, as depicted in Figure 1, the agent is represented by a deep neural network (NN). This neural network makes decisions at each time step (*t*), thereby influencing the environment's state and receiving feedback in the form of a reward that assesses the effectiveness of the chosen action. Over multiple iterations, the agent improves its decision-making abilities by selecting actions based on the policy and learning from accumulated states, actions, and rewards. The ultimate objective is to determine an optimal policy (π*(*a*|*s*)) that maximizes the long-term reward.

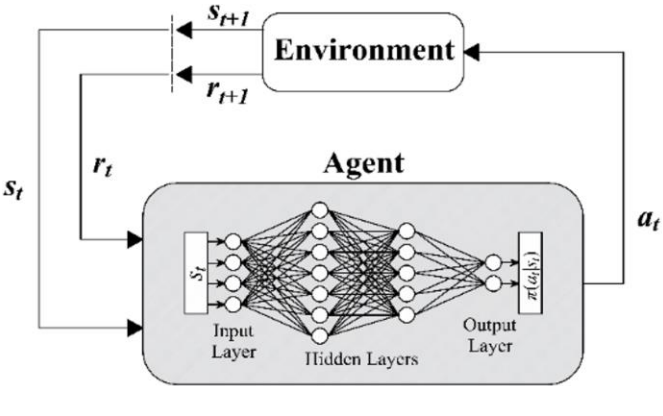


Figure 1. The DRL concept representation

2.2 Transformer model

A neural network architecture called the transformer model is frequently utilized for tasks involving natural language processing, but it may also be employed for other sequential data analysis jobs including time series modelling. It is a good option for assessing and managing vibrations in active mass dampers under seismic conditions because of its capacity to record intricate patterns and dependencies in sequential data.

Designed with a specific focus on sequential data, such as time series and text, the transformer model stands as a robust neural network architecture with significant capabilities. It has been engineered to effectively process and analyze such data, showcasing its power and versatility in various applications. It overcomes some limitations of traditional recurrent neural networks (RNNs), such as long short-term memory (LSTM). One key aspect of the transformer model is its ability to capture long-range dependencies in the data. Traditional sequential models like LSTM often struggle with maintaining information over long sequences, which can hinder their effectiveness. The transformer model addresses this by incorporating a concept called self-attention. Self-attention allows the model to calculate and represent the relationships between different elements in the input sequence without relying on the sequential order of the data. This means that the model can understand the connections and dependencies between elements regardless of their position in the sequence. This is particularly useful for capturing complex patterns and relationships in sequential data. LSTM [18] stands out as the most successful artificial neural network for managing sequential data and time series modeling, representing a variant of recurrent neural network (RNN). Its proficiency extends to capturing the temporal evolution in various contexts. Despite its capability to address traditional RNN limitations like gradient vanishing and explosion [19], LSTM often exhibits slowness in training due to its sequential data introduction requirement, hindering parallelization and necessitating GPU usage. Additionally, it has displayed constraints in handling long-range dependencies.

To overcome these challenges, the transformer [20] was introduced, incorporating the self-attention concept to calculate and represent the input and output data without the need to feed sequential data. In this study, the transformer model was used to show the time evolution in the acceleration fields represented in the turbulent flow data.The Transformer model in this study, depicted in Figure 2, mirrors the original Transformer and comprises two primary components: the encoder and the decoder. Both components process inputs through positional encoding (PE), employing sine and cosine functions to encode order position information into a vector inserted straight into the vector of input.

Six stacked encoder layers make up the encoder, and each one has a feed-forward and a multi-head self-attention sub-layer. In the multi-head self-attention sub-layer, the input includes queries, keys, and values, and attention serves as a mapping function that calculates the weighted sum of values based on queries and key-value pairs.

Scaled dot-product attention, an attention mechanism in which the dot products are scaled down by $\sqrt{d_k}$, is used to illustrate the attention function. The scaled dot-product attention is calculated as:

$$Attention(Q,K,V) = soft\max(\frac{QK^T}{\sqrt{d_K}})V \tag{1}$$

where, Q, K, V are matrices that contain the queries, keys, and values, respectively. d_k is the dimension of Q, K. Here $d_k = d_v = d_{model}/h$, where d_v, d_{model}, and h are the dimension of V, the dimension of the input data to the model, and the number of heads, respectively.

The multi-head self-attention sub-layer in the transformer model enables the model to collectively focus on and process information from diverse representation subspaces at various positions. This mechanism facilitates the model's ability to capture and integrate contextual dependencies across the input sequence, leading to a comprehensive understanding and analysis of the sequential data.

$$r_t = -[(D_t^2)_{1^{st}} + (D_t^2)_{2^{nd}} + (D_t^2)_{3^{rd}}] \tag{2}$$

$$head_i = Attention(QW_i^Q, KW_i^k, VW_i^v) \tag{3}$$

where, $W_i^Q . W_i^K . W_i^V .$ and W^o are the weights corresponding to Q, K, V at every head, respectively; W^o represents the weights of the concatenated heads. $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}, W_i^K \in \mathbb{R}^{d_{model} \times d_k}, W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, and $W^O \in \mathbb{R}^{hd_v \times d_{model}}$.

The feed-forward sub-layer of the encoder layer is made up of two dense layers with ReLU and linear activation functions. Before performing layer normalisation, a residual connection is used, same like in the multi-head self-attention sub-layer. With the use of this connection, the vector can be projected into a higher-dimensional space and then projected back to its original dimension, making it easier to extract pertinent information. Similar to the encoder, the decoder layer is made up of six layers. The decoder layer includes a third sub-layer that handles multi-head attention over the encoder stack's output in addition to the feed-forward and multi-head self-attention sub-layers. With the help of this attention mechanism, the decoder can efficiently extract information from the encoder's output, which facilitates the creation of precise predictions. Moreover, as shown in Figure 2, the multi-head self-attention sub-layer in the decoder is transformed into a masked multi-head self-attention sub-layer. This modification involves applying a mask to the attention mechanism, ensuring that during training, the decoder attends only to the available information at each decoding step. This prevents the model from accessing future positions, maintaining the integrity of the training process. Overall, the transformer model utilizes the feed-forward and multi-head self-attention sub-layers in the encoder and decoder layers to process and extract important information from the input sequence. The decoder layer includes additional multi-head attention over the encoder stack's output, enhancing prediction accuracy. The masked multi-head self-attention sub-layer in the decoder ensures that predictions are solely based on the available information, preventing any influence from future positions.jh This resembles the multi-head self-attention sub-layer except that the scaled dot-product attention is modified to a masked scaled dot-product attention [20]. In order to prevent eventual information leaking, the masking procedure makes sure that the prediction can only rely on the known outputs. In this work, the rate of dropout is fixed at 0.1 and the dropout approach is applied to each sub-layer prior to the residual connection. The Transformer model's loss function is determined to be the square of the L_2 norm error, such that:

$$l_{Transformer} = \frac{1}{M}\sum_{m=1}^{M} \left\| output_m - T\arg et_m \right\|_2^2 \tag{4}$$

When, at a given time step, m, Output and Target stand for the ground truth and the output of the Transformer model, respectively.The optimization algorithm known as adaptive moment estimation (Adam) [21] is employed in the weights' updating. Mini-batches are created from the training data, with a 64-bit size limit for each minibatch. Using random actions, the transformer model is utilised in this work to forecast the temporal evolution of the displacement of each level in the three-story building. Here, the input to the model is the instantaneous actions with the accelerations and displacements and the output is the predicted values of a specific future time step. More details are shown in the next section.

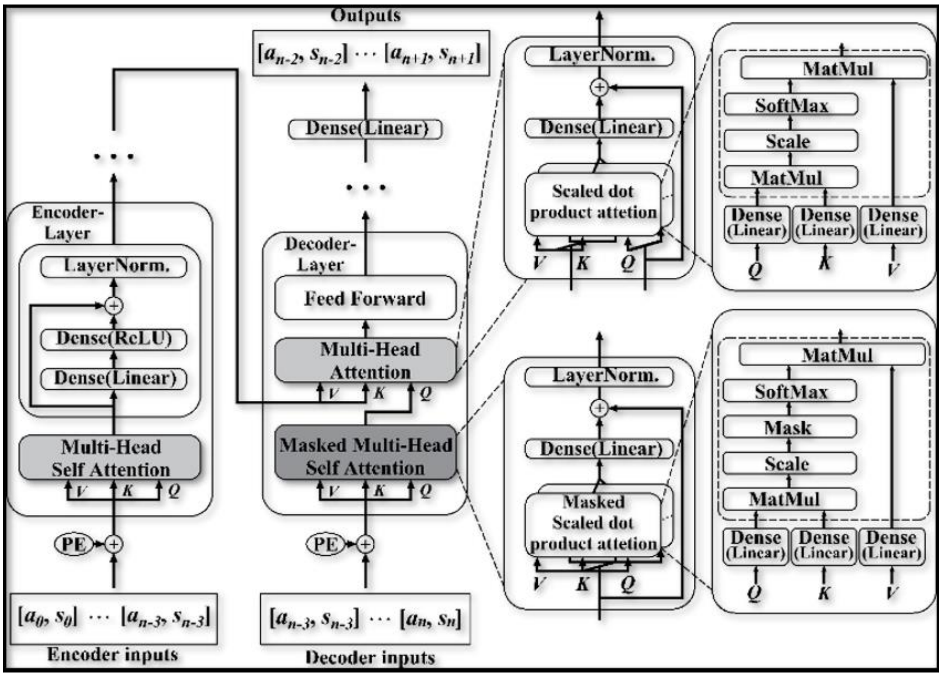

Outputs
[a_{n-2}, s_{n-2}] ··· [a_{n+1}, s_{n+1}]
Dense(Linear)
Encoder-Layer
LayerNorm.
Dense(ReLU)
Dense(Linear)
Multi-Head Self Attention
V K Q
PE
[a_0, s_0] ··· [a_{n-3}, s_{n-3}]
Encoder inputs
Decoder-Layer
Feed Forward
Multi-Head Attention
Masked Multi-Head Self Attention
[a_{n-3}, s_{n-3}] ··· [a_n, s_n]
Decoder inputs
LayerNorm.
Dense(Linear)
Scaled dot product attetion
Masked Scaled dot product attetion
MatMul
SoftMax
Scale
Mask
Dense (Linear)
Q K V

Figure 2. The construction of the transformer

2.3 DRL framework

This work utilizes a DRL model based on PPO, which falls within the family of policy gradient methods [22]. PPO is commonly employed in DRL-based control problems due to its simplicity, faster execution compared to Trust Region Policy Optimization (TRPO) [23], and minimal need for meta-parameter tuning. It is particularly well-suited for continuous control problems and outperforms DQN. Appendix provides a brief introduction to the PPO method. In this study, the objective is to develop a model that optimally adjusts the control signal of AMD (Active Mass Damper), represented by voltage values as depicted in Figure 3. The instantaneous values, denoted as a_t, are added to the control signal u_t derived from a classical LQR model at each iteration step. The agent operates under a policy $\pi(a_t|s_t)$, where $a_{i,j}^n$ belongs to the action range [-1,1] multiplied by u_{rms} (root mean square of the control signal from the LQR model). At each step, the agent receives the next state s_{t+1} and reward r_{t+1} from the environment based on the action a_t taken. The states are defined by the accelerations and displacements of the three floors. The reward function incorporates the drifts of the three levels, enabling the DRL model to determine the most favorable action to take. Therefore, the DRL model's goal is to discover the control strategy that maximizes the anticipated long-term rewards [24]:

$$\pi^* = \arg\max E_\pi\left(\sum_{t=1}^{N} \gamma^{(t-1)} r_t\right) \tag{5}$$

where, $\gamma^{(t-1)}$ is the (t-1)th power of the discount factor γ, which estimate the weight of the immediate rewards in the future steps of iteration . In this work, the term of γ is set to 0.9. The immediate reward function is defined as:

$$r_t = -[(D_t^2)_{1^{st}} + (D_t^2)_{2^{nd}} + (D_t^2)_{3^{rd}}] \tag{6}$$

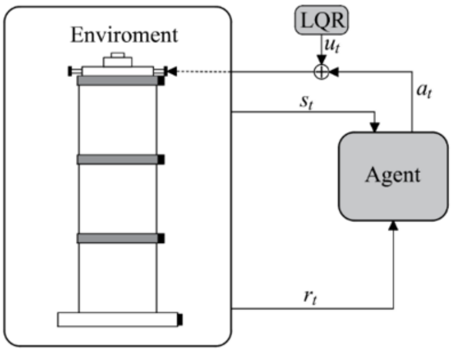


Figure 3. Diagram for framework the building structure's active vibration control with DRL

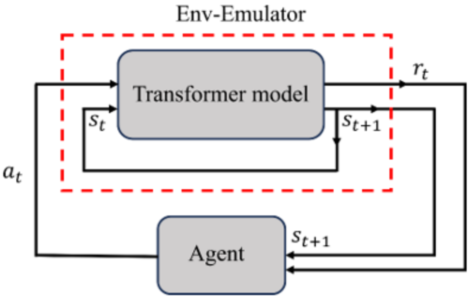


Figure 4. Training process with transformer

For every episode in our investigation, we employ 50 iteration steps to train the DRL model. As previously stated, this study utilizes the transformer model to forecast the temporal progression of each floor's displacement in the building, serving as an emulator in the training phase. As shown in Figure 4, the model takes instantaneous actions along with accelerations and displacements as input and generates predicted values for a specific future time step. The structure of the DRL model used in this study is implemented using the tensorforce library [25].

3. EXPERIMENTAL SETUP AND EARTHQUAKE SIMULATION

The practical test was conducted by designing and manufacturing a three-story shear building structure model with dimensions of (50×40×40) cm for each floor. The floor surface was made of wood while the columns were made of steel, taking into account the addition of damping factors between the floors. The weight of the floor is 45 Newtons, and AMD was installed at the top of the structural model on the roof of the third floor, which consists of a 42HD4027 NEMA17 Stepper Motor, in addition to the active mass that weighs 7 Newtons, equivalent to 5% of the total weight of the structural model. The optimal performance was evaluated by conducting a test on the Shaking tables are commonly utilized in earthquake simulation scenarios, wherein a tabletop is fixed atop a series of mechanical vibrators or actuators. The purpose of the table surface is to simulate the motion and vibrations that occur during an earthquake in order to mimic the behavior of the ground. A specimen or scale model of a structure is set up on the shaking table for the experiment. The table is then programmed to replicate the collected ground acceleration data, so producing distinct earthquake motions. A vast variety of ground motions, including various frequencies, amplitudes, and waveforms, can be replicated using the shaking table. Shaking tables are widely used in structural testing, earthquake engineering research, and seismic qualifying of buildings. the shaking table that was manufactured in the laboratory by connecting the structure to a moving base using a 42HD4027 NEMA17 Stepper Motor, in addition to a 240 watt power supply with a high power capacity of 20 kN and max. Stroke 200 mm . The Arduino Mega 2560 microcontroller board is used to control the actuator with TB6600 stepper driver module for stepper motor in AMD and shaking table when using the time hestory for acceleration ampltuted EL centro(scaled version) [26] by apply tha data for it to generation the vibration in the model strcture by shaking table, in addition to using an SD card adapter to convert the analog signal as shown in the Figure 5.

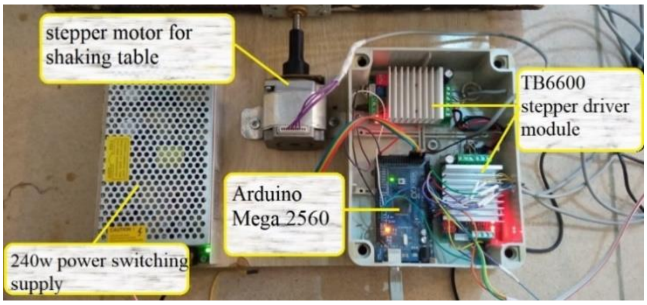

stepper motor for shaking table
TB6600 stepper driver module
Arduino Mega 2560
240w power switching supply

Figure 5. The components of experimental work

Four ADXL335 accelerometers were installed and distributed on the floors, in addition to installing one of them on the AMD To calculate the absolute acceleration. The measurement range for the acceleration was ±25 m/s^2. The control force of the AMD was obtained through the product of the acceleration multiplied by the active mass. At the same time, distance sensors were used. VL53L0X is distributed to all floors of the structural model in order to measure the relative displacement of each floor,see Figure 6. The AMD controller was implemented by building the DRL-LQR controller on the host desktop using MATLAB/Simulink and then connecting it to the Arduino Mega 2560 I/O using blocks provided by Simulink Real-Time. The DRL-LQR needed structural states to control the feedback, but these were not observable in testing. Therefore, the control force was calculated using the recorded acceleration as input. However, noise in the accelerometer measurement may cause high-frequency control force and increase the acceleration response. As a result, a fourth-order low-pass filter with a cut-off frequency of 20 Hz was created.

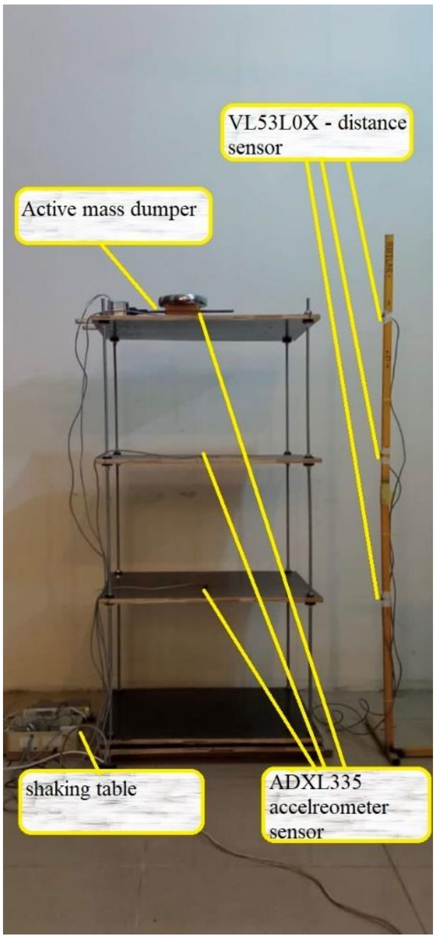


Figure 6. The components of experimental work

When compared to simulation-based research, implementing a DRL controller in an experimental context can have a number of drawbacks and difficulties. Firstly, the Experimental Setup and HardwareActuators and sensors are examples of the physical hardware needed in an experimental context to interact with the structure and gather data. Hardware component design and setup can be difficult and time-consuming. To guarantee precise measurements and appropriate control action, the hardware required meticulous calibration and synchronization. Real-time control is another important component of the experimental setup. In this scenario, the DRL controller must function in real-time to deliver control inputs based on the collected sensor data. Because real-time implementation requires that the DRL algorithms be completed within the allotted time, it may create limits on their computational complexity. To meet the demands of real-time control, hardware resources, and efficient algorithms are needed. so that the transformer concept was employed.

4. RESULTS AND DISCUSSION

First and foremost, the study delves into the learning process of the proposed model. Illustrated in Figure 7, the reward values exhibit a gradual increase, plateauing at a consistent level after just 600 episodes, underscoring the commendable learning performance of the model. The 600 training episodes are considered sufficient, and this is evident from the stability of the system's reward values, which indicates that the system has reached the highest possible stable state within the available controller, which in turn depends on stable policy noting that the reward values become almost constant after the 600 training episodes. Notice Figure 7 at the Training 800 episodes the reward value equals approximal -30 for smoothed reword which is the same value for 600 training episodes for smoothed reword.

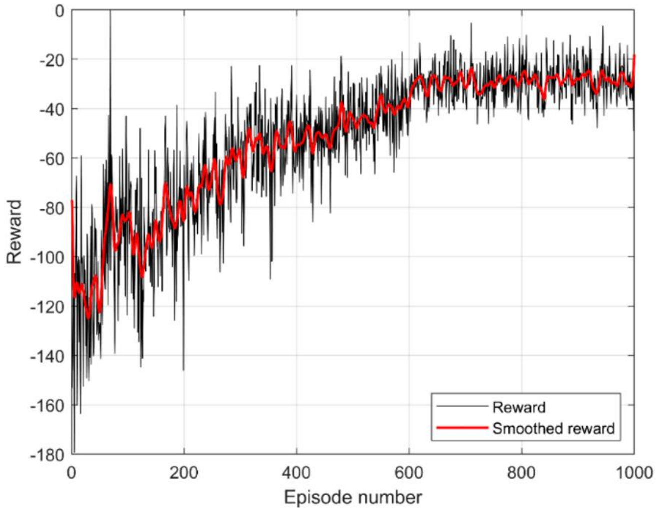


Figure 7. Reward function evaluation during the training process (The red plot represents the smoothed reward values)

Notably, there is no real-time interaction between the DRL agent and the real experimental environment during the training phase, which leads to a significant decrease in the computing cost, symbolised by the training time. Analysing the control signal's immediate effect on the three floors' displacements and accelerations reveals some intriguing trends. Figure 6 illustrates a tendency for displacement values to increase with the increase in floor levels. The DRL model's results, depicted in Figure 8, show fewer values compared to the no-control case, suggesting the DRL model effectively mitigates floor displacement. Furthermore, Figure 9. showcases the model's remarkable capability to reduce the acceleration of each floor consistently throughout the artificial ground excitation period, maintaining its performance across all floors during this period. The LQR controller offers a baseline control technique that guarantees stability and optimal control by integrating DRL with LQR control. Subsequently, DRL refines the control strategies that the LQR controller has learned, resulting in enhanced structural responsiveness and vibration reduction performance. The DRL training procedure begins with the original LQR controller. It cuts down on the amount of training time needed to get optimal control by offering a stable control policy that enables DRL to converge more quickly. To show the effectiveness of this type of controller comparisons are provided between the DRL and uncontrolled cases, In addition to comparing it with the most ideal case applied throughout this article, DRL-LQR can be represented in literature [27].

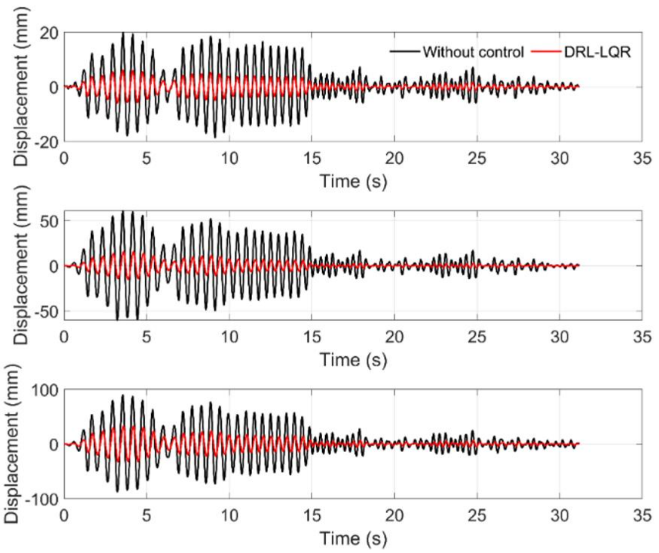


Figure 8. Plots showing the first floor (top), second floor (middle), and third floor (bottom) in real time

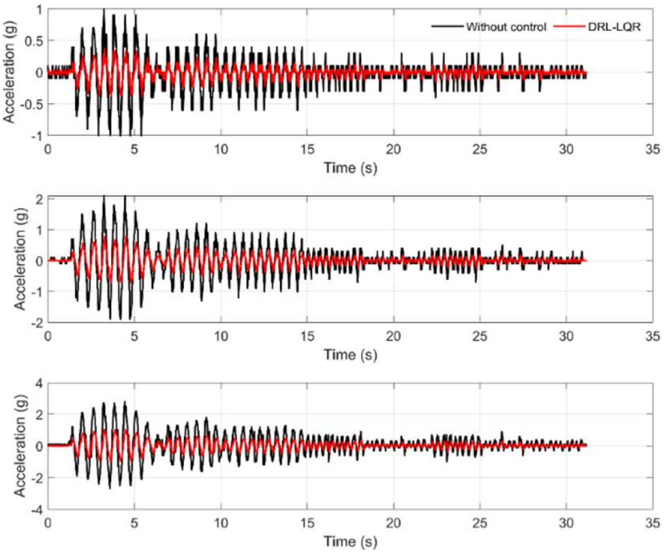
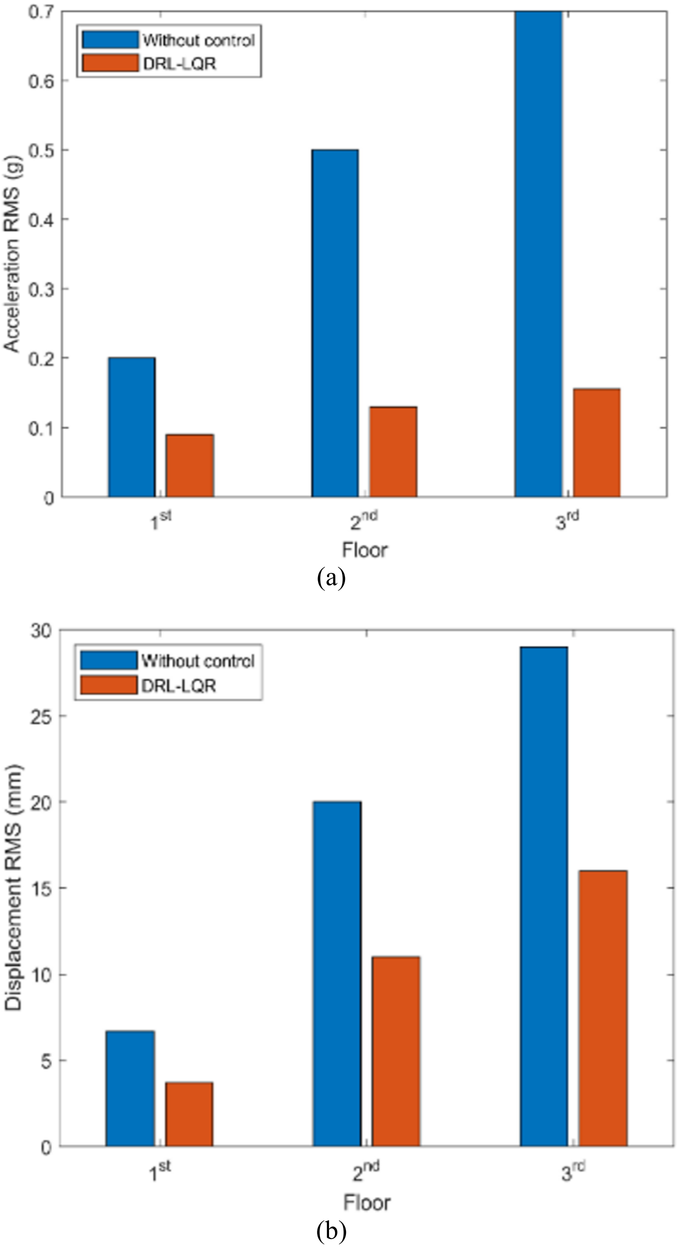


Figure 9. Plots showing the first floor's instantaneous acceleration at the top, middle, and bottom

The root mean square (RMS) of the accelerations and displacements on the three floors is measured as part of the statistical assessment of the model's performance. As Figure 10(a) illustrates, the RMS of floor accelerations is significantly lower than in the no-control case, by about 50–70%. In a similar vein, the model's performance is shown in terms of the RMS of floor displacements in Figure 10(b).The RMS acceleration and displacement reductions compared to the uncontrolled case for experimental tests without control,that is means the time history for the EL Centro earthquake as scaled version applied for the model structure by shaking table without applying any control for AMD that is fixing in the top floor of the structure (without DRL-LQR) and extracting experimental results related to RMS of displacement or acceleration. It is evident that the model has learned to adjust signals from the classical LQR model, striving to achieve optimal reductions in both floor displacement and acceleration. Figure11 provides an overarching view of AMD's behavior under the control of the DRL model. In Figure11(a), the AMD's acceleration behavior aligns relatively consistently with the building during the excitation period, although the acceleration values of the AMD exhibit a more randomized pattern. This randomness is linked to the DRL model's pursuit of optimal performance, adapting to the dynamic structure of the model with quasi-random changes in the AMD position, as reflected in Figure 9.

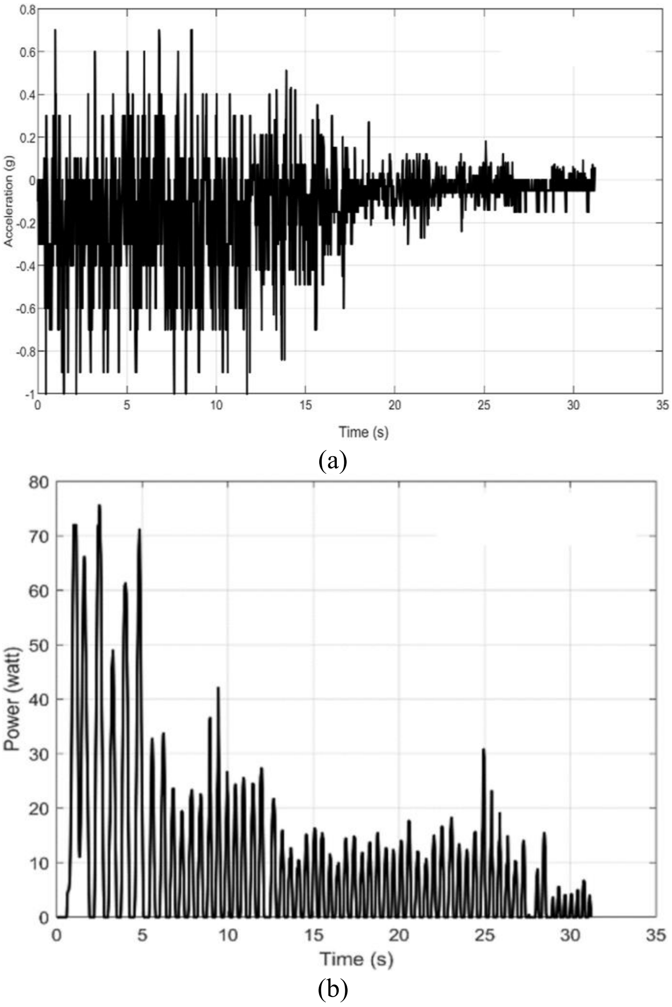


(a)

(b)

Figure 10. (a) RMS of each floor's acceleration; (b) RMS of each floor's displacement

Furthermore, Figure 11(b) illustrates a direct correlation between the consumed power of the AMD and the amplitude of the simulated earthquake. Notably, the AMD's response displays less random behavior compared to its acceleration values. This correlation emphasizes the model's ability to efficiently modulate the AMD's power consumption in response to varying earthquake conditions. The interplay between AMD behavior, DRL model control, and earthquake dynamics, as visualized in Figure 9, underscores the model's capability to intricately manage the AMD's response in a way that aligns with the building's characteristics and seismic inputs. It should be noted that All details related to the comparison of results of performance between the case of DRL and LQR controller were included Severally within the citation from the article mentioned [27]. Lastly , The main goal of the article was to shed light on the efficiency of the controller experimentally in reducing the displacement and acceleration of the structure mainly within the experimental test of the laboratory structure model. As for the computational cost/training time for the new controller DRL-LQR and the previous DRL, it was included to indicate that increasing the efficiency in it was relied upon theoretically in previous research has been included in the previous references



(a)

(b)

Figure 11. Instantaneous AMD acceleration (a) and instantaneous AMD power (b)

5. CONCLUSIONS

A recently created transformer-based DRL framework for active building structure vibration control under seismic effects is presented in this work. The framework creates control actions that improve the performance of a traditional LQR model by fusing ANNs with the PPO method. Furthermore, an emulator model in the form of a transformer model is employed to replicate the training environment. The training outcomes show how successful the suggested structure is. The model demonstrates the capacity to learn in a very short amount of episodes, leading to a notable decrease in training expenses. The instantaneous results of the displacements and accelerations of the structure model floors exhibit a remarkable reduction, surpassing the performance of the LQR model. Furthermore, the statistical results indicate a 50 to 70% improvement in the reduction of drifts compared to the no-control case. The instantaneous displacement and acceleration findings for the structure model floors show a significant decrease in the LQR model's performance. Additionally, compared to the no-control condition, the statistical results show a 50–70% improvement in the drift decrease. The AMD response, power, and building accelerations are correlated, and this indicates that the DRL model learns to generate the best control signals without any prior understanding of the structure's dynamics. This highlights the capability of the transformer-based DRL framework to adapt and optimize control actions based on the observed structural response. The findings of this study suggest that the combination of transformer models and DRL has significant potential for practical applications in active vibration control, including multi-story building structures. The framework not only achieves remarkable reductions in training time but also offers promising results in terms of control performance. However, it is important to acknowledge the limitations of this study. The practical implementation of the proposed framework on larger-scale structures may encounter challenges related to hardware requirements, control loop design, system identification, and data acquisition strategies. These factors need to be carefully considered and addressed in future research. Future research directions can focus on enhancing the transformer-based DRL framework by exploring advanced DRL algorithms that can efficiently handle larger-scale systems while maintaining real-time control capabilities. Additionally, investigating hybrid control strategies that combine DRL with other control techniques could further enhance performance, improve robustness, and reduce computational requirements. Furthermore, experimental validation on larger and more complex structures will be crucial to validate and extend the findings of this study.

REFERENCES

[1] Spencer Jr, B.F., Nagarajaiah, S. (2003). State of the art of structural control. Journal of Structural Engineering, 129(7): 845-856. https://doi.org/10.1061/(ASCE)0733-9445(2003)129:7(845)

[2] Cheng, F.Y., Jiang, H., Lou, K. (2008). Smart Structures: Innovative Systems for Seismic Response Control. CRC Press. https://doi.org/10.1201/9781420008173

[3] Dyke, S.J., Spencer Jr, B.F., Quast, P., Kaspari Jr, D.C., Sain, M. K. (1996). Implementation of an active mass driver using acceleration feedback control. Computer-Aided Civil and Infrastructure Engineering, 11(5): 305-323. https://doi.org/10.1111/j.1467-8667.1996.tb00445.x

[4] Wang, L., Nagarajaiah, S., Shi, W., Zhou, Y. (2022). Seismic performance improvement of base-isolated structures using a semi-active tuned mass damper. Engineering Structures, 271: 114963. https://doi.org/10.1016/j.engstruct.2022.114963

[5] Bossens, F., Preumont, A. (2001). Active tendon control of cable-stayed bridges: A large-scale demonstration. Earthquake Engineering & Structural Dynamics, 30(7): 961-979. https://doi.org/10.1002/eqe.40

[6] Bagherkhani, A., Baghlani, A. (2021). Reliability assessment of MR fluid dampers in passive and semi-active seismic control of structures. Probabilistic Engineering Mechanics, 63: 103114. https://doi.org/10.1016/j.probengmech.2020.103114

[7] Karami, K., Manie, S., Ghafouri, K., Nagarajaiah, S. (2019). Nonlinear structural control using integrated DDA/ISMP and semi-active tuned mass damper. Engineering Structures, 181: 589-604. https://doi.org/10.1016/j.engstruct.2018.12.059

[8] Soto, M.G., Adeli, H. (2019). Semi-active vibration control of smart isolated highway bridge structures using replicator dynamics. Engineering structures, 186: 536-552. 10.1016/j.engstruct.2019.02.031

[9] Reinhorn, A.M., Soong, T.T., Lin, R.C., Riley, M.A., Wang, Y.P., Aizawa, S., Higashino, M. (1992). Active bracing system: A full scale implementation of active control. National Center for Earthquake Engineering Research.

[10] Yang, J.N. (1975). Application of optimal control theory to civil engineering structures. Journal of the Engineering Mechanics Division, 101(6): 819-838. https://doi.org/10.1061/JMCEA3.0002075

[11] Song, G., Gu, H. (2007). Active vibration suppression of a smart flexible beam using a sliding mode based controller. Journal of Vibration and Control, 13(8): 1095-1107. https://doi.org/10.1177/1077546307078752

[12] Casciati, F., Rodellar, J., Yildirim, U. (2012). Active and semi-active control of structures-theory and applications: A review of recent advances. Journal of Intelligent Material Systems and Structures, 23(11): 1181-1195. https://doi.org/10.1177/1045389X12445029

[13] Ying, Z.G., Ni, Y.Q. (2015). Optimal control for vibration peak reduction via minimizing large responses. Structural Control and Health Monitoring, 22(5): 826-846. https://doi.org/10.1002/stc.1722

[14] Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep Learning. MIT Press.

[15] Rahmani, H.R., Chase, G., Wiering, M., Könke, C. (2019). A framework for brain learning-based control of smart structures. Advanced Engineering Informatics, 42: 100986. https://doi.org/10.1016/j.aei.2019.100986

[16] Kim, H.S., Kim, U. (2023). Development of a control algorithm for a semi-active mid-story isolation system using reinforcement learning. Applied Sciences, 13(4): 2053. https://doi.org/10.3390/app13042053

[17] Zhang, Y.A., Zhu, S. (2023). Novel model-free optimal active vibration control strategy based on deep reinforcement learning. Structural Control and Health Monitoring, 2023: 6770137. https://doi.org/10.1155/2023/6770137

[18] Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8): 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

[19] Graves, A., Fernández, S., Schmidhuber, J. (2005). Bidirectional LSTM networks for improved phoneme classification and recognition. In International Conference on Artificial Neural Networks, Warsaw, Poland, pp. 799-804. https://doi.org/10.1007/11550907_126

[20] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems.

[21] Kingma, D.P., Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. https://doi.org/10.48550/arXiv.1412.6980

[22] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347. https://doi.org/10.48550/arXiv.1707.06347

[23] Schulman, J., Levine, S., Abbeel, P., Jordan, M., Moritz, P. (2015). Trust region policy optimization. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, pp. 1889-1897.

[24] Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540): 529-533. https://doi.org/10.1038/nature14236

[25] Kuhnle, A., Schaarschmidt, M., Fricke, K., (2017). Tensor force: A TensorFlow library for applied reinforcement learning. https://github.com/tensorforce/tensorforce.

[26] Gudarzi, M., Zamanian, H. (2013). Application of active vibration control for earthquake protection of multi-structural buildings. International Journal of Scientific Research in Knowledge, 1(11): 502-513.

[27] Gheni, E.Z., Al-Khafaji, H.M., Alwan, H.M. (2024). A deep reinforcement learning framework to modify LQR for an active vibration control applied to 2D building models. Open Engineering, 14(1): 20220496.

APPENDIX

PPO algorithm

Recognized for its stability and proficiency in handling continuous action spaces, Proximal Policy Optimization (PPO) is particularly well-suited for tasks involving continuous control actions. PPO operates within the framework of policy gradient algorithms, aiming to directly derive the optimal policy, denoted as $\pi^*(a_t|s_t)$, which maximizes the long-term reward function, $R(t) = \sum_{t=1} \gamma^{t-1} r_t$, where γ is the discount factor and its range is between 0 and 1. Unlike in other methods such as Q-learning, where an indirect description of the policy is represented by the artificial neural network (ANN), in the policy gradient methods, the policy is directly obtained by utilizing the ANN. The goal of training in the policy gradient methods is the obtain the maximum reward such that:

$$R_{max} = \underset{\Theta}{\max} \mathbb{E}\left[\sum_{t=0}^{H} R(s_t)|\,\pi_\Theta\right] \tag{A.1}$$

where, π_Θ is the policy function, Θ represents the weights of the ANN, and s_t represents the state of the system.

If we denote τ as a (*s, a, r*)-based sequence,

$$\tau = (s_0, a_0, r_0), (s_1, a_1, r_1), \dots, (s_H, a_H, r_H) \tag{A.2}$$

Then, we can define a value function (which is the quantity that should be maximized) as,

$$\nabla_\Theta = \mathbb{E}\left[\sum_{t=0}^{H} R(s_t, u_t)|\,\pi_\Theta\right] = \sum_\tau \mathrm{P}(\tau, \Theta) R(\tau) \tag{A.3}$$

With mathematical manipulations, one can obtain,

$$\begin{aligned} \nabla_\Theta V(\Theta) &= \sum_\tau \nabla_\Theta \mathrm{P}(\tau, \Theta) R(\tau) \\ &= \sum_\tau \frac{\mathrm{P}(\tau, \Theta)}{\mathrm{P}(\tau, \Theta)} \nabla_\Theta \mathrm{P}(\tau, \Theta) R(\tau) \\ &= \sum_\tau \mathrm{P}(\tau, \Theta) \frac{\nabla_\Theta \mathrm{P}(\tau, \Theta)}{\mathrm{P}(\tau, \Theta)} R(\tau) \\ &= \sum_\tau \mathrm{P}(\tau, \Theta) \nabla_\Theta \log\big(\mathrm{P}(\tau, \Theta)\big) R(\tau) \end{aligned} \tag{A.4}$$

Eq. (A.4) represents a new expected value, which can be sampled under π_Θ and used as the input to the gradient descent. Here one can estimate the policy-dependent log-prob gradient as:

$$\begin{aligned} &\nabla_\Theta \log\left(\mathrm{P}(\tau^i, \Theta)\right) \\ &= \nabla_\Theta \log\left[\prod_t \mathrm{P}\big(s_{t+1}^i|s_t^i, a_t^i\big) \pi_\Theta\big(a_t^i|s_t^i\big)\right] \\ &= \nabla_\Theta \left[\log \mathrm{P}\big(s_{t+1}^i|s_t^i, a_t^i\big) + \sum_t \log \pi_\Theta\big(a_t^i|s_t^i\big)\right] \\ &= \nabla_\Theta \sum_t \log \pi_\Theta\big(a_t^i|s_t^i\big) \end{aligned} \tag{A.5}$$