Balancing a Cart Pole Using Reinforcement Learning in OpenAI Gym Environment

Avneesh Narwal
Department of Computer Science
Lovely Proffesional University

Abstract— **Support Learning (RL) could be a subcategory of machine learning. The extraordinary highlight of fortification learning that recognizes it from other machine learning approaches is the self-training of the specialist from gotten data and input from the environment. The suitable activity choice guided the operator towards the way better ideal arrangement. The operator has no prior knowledge almost the environment, the specialist must investigate each perspective of the environment based on input. This foremost advantage of RL calculations suits complex ideal control issues such as a Cart post issue modified pendulum issue and mechanical technology where no earlier data on the framework flow is accessible. In this paper, the conventional mechanical Cartpole framework is controlled by utilizing Q-learning models, and for assessment measures, Cruel Square mistake (MSE) and Cruel Outright Mistake (MAE) are connected in calculations with the presentation of OpenAI Exercise center environment.**

Keywords— Reinforcement Learning, Cart Pole, OpenAI Gym, Q Learning

I. Introduction

Reinforcement Learning is a training method of machine learning involving reward and penalty for desired behavior and for undesired behavior respectively. The Reinforcement Learning agent through consciousness interprets its environment, takes actions, and modifies its learning paradigms about its environment.

The agent selects a particular action according to the interaction with the robust and dynamic environment. Such robust and dynamic controllers are widely used in real-world problems as PID controllers and fuzzy controllers where frequent adjustments are required for efficient performance of applications. When RL is applied to such a robust system, the agent must interact with the unknown environment and try to achieve maximum cumulative reward [2]. The traditional methods for these mechanical systems are made up of existing physics-based concepts and mathematical formulations. But these traditional procedures are executed manually by twisting control parameters which induces various issues and errors in the execution of mechanical systems. [1, 2].

According to the learning process, methods, and applications, machine learning algorithms are divided into supervised learning, unsupervised learning, semi-supervised, and reinforcement learning. In supervised machine learning, the mapping between some input data and output data is already available, known as labeled data and by using these predefined labeled data the machine trains itself and then predicts output for future input values.

The supervised learning algorithm is suitable for classification and regression problems. While in unsupervised learning, the algorithm tries to discover patterns between unlabeled datasets given as input. [13] Unsupervised learning applied to clustering and association problems.

Reinforcement Learning (RL) executes through the concept of the feedback process, where the agent must interact with the environment and perform the action on the particular state for optimal reward function according to requirements. This process in the loop is a trial-and-error procedure where the agent decides which particular action on that specific state helps achieve the target state in the system.

After performing action on the current state next state is generated by the environment and as reward feedback is returned to the agent. After performing this procedure in the loop, the agent learns the best action on the state for achieving a maximum cumulative reward [11].

Mostly mechanical and underactuated systems are very suitable for reinforcement learning-based research areas due to their dynamic and complex nature. In Reinforcement Learning, the agent evaluates each aspect of real-time systems to ensure its optimal performance. So, for the real time systems reinforcement learning algorithms such as Q-learning and deep Q-learning are very popular among researchers. The proficiency of RL has impelled towards the various domains and solved the domain-specific challenges efficiently. The applications RL are mostly in these fields' robotics, game playing, self-driving cars, and resource management. Drug discovery and financial trading. In robotics, RL trains the agent to learn complex tasks such as grasping objects, and obstacle detection. Similarly, in the gaming areas, the RL agents got expertise like human experts. For autonomous driving, the RL agent makes real-time decisions based on traffic navigation scenarios and takes decisions according. Another application of RL is resource management, The RL agent is responsible for optimizing the resources as inventory management, traffic navigation, and improving efficiency by optimizing energy distribution. Similarly. RL trained the agent for identifying potential drug candidates and analyzing market data and based on these optimized data performed the strategy for maximum returns.

Reinforcement Learning computes the optimal solution which is the maximum result in the minimum time for complex and dynamic problem domains. The agent

understands the environment after performing the same procedure repeatedly and training its knowledge about the environment.

In previous research work, several studies were performed to solve the traditional control problem of cart pole balancing.
However, most studies considered the theoretical aspects of physics for solving the cart pole problem. So, for further research, reinforcement learning approaches attract the attention of the researcher for such physics-based problems. In this research work, the concept of Q learning is proposed with two reward functions mean square error (MSE) and Mean Absolute error (MAE). The fast and stable convergence of the cart pole balancing problem obtained by Q-learning with the MSE and MAE as reward function evaluated and performed more efficiently than traditional physics-based concepts.

The following sections are about the research problem - the cart pole balancing, an overview of Reinforcement Learning, OpenAI Gym and Q Learning, and reward functions. The cart pole system is defined in Section II in detail, and Section III is about RL concepts. OpenAI Gym explained in Section IV and Section V defined different reward functions and the conclusion in Section VI.

II. The Cart-Pole Balancing Problem

By using the Reinforcement learning algorithms, the RL agent is trained to balance the pole joint with the cart at some pivot point which moves horizontally on the surface. The cart pole system mainly has two components a simple cart and a vertical bar. The pole on the cart was fixed at some pivot point and the cart can move in left or write directions as explained in Fig 1.

In the Cart Pole Environment, the agent explores all the possible actions and their corresponding rewards. Then, update its policy towards optimal reward achievement. The goal of this problem is to find a control policy for balancing the pole in an upward direction by using bidirectional force applied to the cart[12].

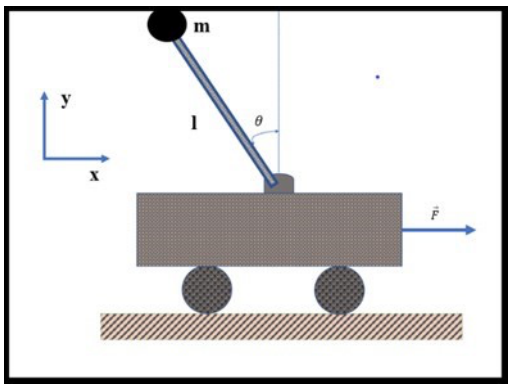

Fig. 1. Cart Pole Dynamics and Control Parameters (Adapted from [1][3])

In Fig 1, the dynamic system of the classical cart pole is explained. The cart is moving on a fixed frictionless surface horizontally due to force F, θ is the deviation of the pole from the pivot point [3]. So, for the cart pole system state space parameters are defined with the help of a four-dimensional vector $\{x, \dot{x}, \theta, \dot{\theta}\}$ where x is the horizontal distance traveled by the cart and $\dot{x}$ is the linear acceleration of the cart. The cart pole system's mathematical formulation is defined in Equation 1.

$$(M+m)\ddot{x} + b\dot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = F(t) \quad (1)$$

$$ml\ddot{x}\cos\theta + \frac{4}{3}ml^2\ddot{\theta} - mgl\sin\theta = 0 \quad (2)$$

In the above-defined Equation 1 and Equation 2, x(t) is the distance traveled by the cart pole on a non-friction surface from the centre point, and $\dot{x}$ and $\ddot{x}$ represent the velocity of the cart and acceleration respectively. For the mathematical computation of the angular acceleration of the pole $\ddot{\theta}$ and linear acceleration of cart $\ddot{x}$ formulas defined in Equation 3 and Equation 4 were applied.

$$\ddot{\theta} = \frac{(M+m)g\sin\theta - \cos\theta[F + ml\dot{\theta}^2\sin\theta]}{\left[\frac{4}{3}\right](M+m)l - ml\cos^2\theta} \quad (3)$$

$$\ddot{x} = \frac{[F + ml[\dot{\theta}^2\sin\theta - \ddot{\theta}\cos\theta]]g}{(M+m)} \quad (4)$$

The Cart pole mechanical system is highly dynamic in nature. The agent has to control the input parameters in such a manner that the pendulum is balanced around its center of mass above the moving cart. In the simulation of the cart pole action space is defined as {LEFT, RIGHT}, which means the cart can move horizontally either in Left or Right direction.

The mathematical formulation for state space parameters of nonlinear dynamics of the cart pole mechanism is defined in equation 5.[2]

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \frac{[F + ml[\dot{\theta}^2\sin\theta - \ddot{\theta}\cos\theta]]g}{(M+m)} \\ \dot{\theta} \\ \frac{(M+m)g\sin\theta - \cos\theta[F + ml\dot{\theta}^2\sin\theta]}{\left[\frac{4}{3}\right](M+m)l - ml\cos^2\theta} \end{bmatrix} \quad (5)$$

III. Reinforcement Learning

A. Basic Concepts

Reinforcement learning is a subfield of machine learning that focuses on self-determining decision-making abilities following the time constraint. In Reinforcement Learning (RL), the observation parameters are dependent upon an agent's behavior, which means the agent should be efficient enough to learn and can improve its performance by learning rather than showing false impressions and negative feedback all the time. So, the agent tries to learn about the anonymous environment by following the trial-and-error procedure for learning optimal action state space and achieving its goals.

Reinforcement Learning includes two major entities Agent and Environment and Actions, reward, and observation are their communication channels, as shown in Fig 2. So, the reinforcement learning entities and their communications are as follows:

- *The Agent:* An agent is an entity that interacts with the environment by performing some actions, afterward taking observations, and finally receives rewards as feedback from the environment.

- *The Environment:* The environment refers to the external system through which an agent interacts. The agent communicates with the environment and receives these information rewards generated by the environment, actions performed by the agent on the environment, and observations the agent receives from the environment.
- *Reward:* Reward is a numerical value, received from the environment. Reward values either be positive or negative. The main purpose of reward is to train the agent to achieve the final goal. The
- *Actions:* Actions are moves that can an agent perform in the environment. In the RL, there are two types of actions discrete or continuous actions performed on the environment. The agent chooses the actions according to their current states and tries to learn the most effective actions to achieve their goal in the given environment.
- *Observations:* The observation is information about the environment on that particular time stamp. Observation provides information about the current state of the agent on that particular time stamp, possible action space for that current state, and other environmental information.

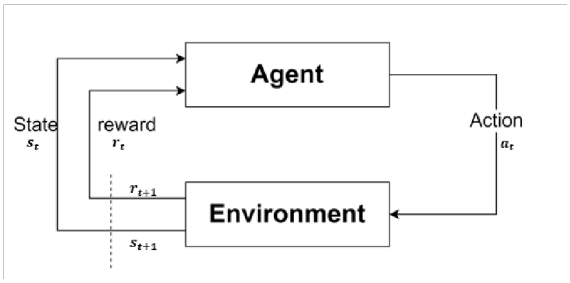

Fig. 2. The reinforcement learning process (Adapted from [4])

The AI agent selects an action from the action space moves toward a new state and receives a reward from the environment as feedback. After repeating these steps, the agent learns which action is best in a particular state to obtain the maximum cumulative reward. As shown in Fig 2, in each iteration the agent receives current states from the environment, then after applying an action the agent's state changes. After repeatedly performing this process, the agent learns from the obtained experience regarding state, action, and corresponding next state and reward. This knowledge helps the agent to achieve the cumulative reward to achieve the goal. The main goal of the Reinforcement learning algorithm is to compute the optimal policy for the given problem. [5, 7]

B. Markov Decision Process (MDP)

The Markov Decision Process (MDP) represents the mathematical framework for dynamic decision-making situations where the performance of the system is influenced by random factors and uncertain system parameters [6]. The MDP framework consists of four key terms such as state S, action A, Transition Probability P, and reward R. So, The MDP for the cart pole problem is $< S,A, P, R, \gamma >$, where-

- State (S): The state space parameters in the cart pole problem represent the current status of the agent which includes cart position, cart velocity, pole angle, and pole angular velocity.
- Action (A): The action set A is about all possible movements that control the dynamics of the cart and pole. In the cart pole environment, the movement of the cart is only possible on the left or right.
- Transition probability (P): P is about the probability distribution given for the current state over the next possible potential successor state.
- Reward (R): Reward R is a numerical value associated with a state-action pair that converges the agent's learning process towards the optimal maximum cumulative sum of reward.
- Discount factor (γ): The discount factor γ determines the influence of the future rewards and determines the preferences for the immediate rewards. The value of the discount factor lies between [0,1].

IV. OpenAI gym and Q-Learning

The OpenAI gym is a standard application programming interface for solving reinforcement learning for environments as classic control and toy text, Atari games, 2D and 3D robots. OpenAI Gym provides the interface for several classical control engineering environments. These interfaces test the efficiency of reinforcement learning so that proposed algorithms can be applied to mechanical systems such as robots, medical fields, etc.

In this paper, For the Cart pole problem, OpenAI Gym is used. In the environment, a pole is attached by a pivot point to a frictionless cart. The pendulum is placed in the upward direction and the cart moves left and right on the surface. In the Cart pole, the agent trying to keep the pole upright. Initially, the pendulum starts from an upward direction and the system aims to prevent the pole from falling after applying force on the cart. The action space of the crat pole is two discrete values (0,1), 0 represents push the cart in the left direction and 1 means push the cart in the right direction according to Figure 3. After performing an action on state, the environment produces an observation state space which consists of cart's position, cart's velocity, the pole angle, and the angular velocity of the pole. The cart position lies between (4.8, 4.8) and the termination condition is (-2.4, 2.4). The pole angle observed between ($\pm 24°$) and episodes terminates when the pole lies outside ($\pm 12°$) range. The +1 reward is assigned for balancing the pole in the upward direction on the cart as long as possible.[8]

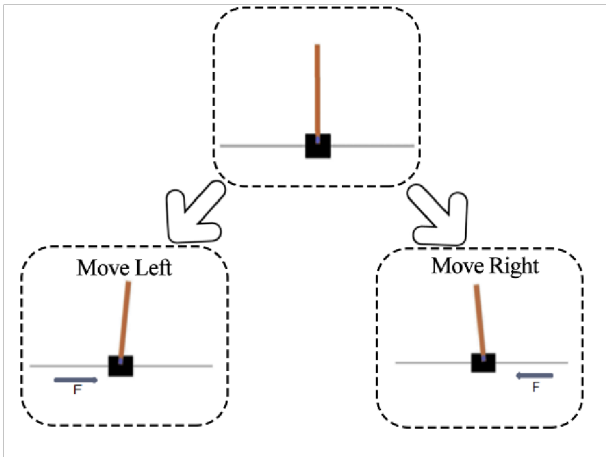

Fig. 3. Action state parameters for Cart pole mechanism [2]

Q-learning is a value-based reinforcement learning algorithm in which the environment is not familiar to the agent and the agent must figure out the best actions for obtaining an optimal solution. In the Q-learning method the samples (S, A, R, S') generated by following policy to maximize $Q(S', A')$ values for achieving the desired target. For the formulation of the Q value, the ε-greedy policy is applied for samples (S, A, R, S') defined in equation (6)

$$Q(S,A) = R(S,A) + \gamma \max_{A} Q(S',A) \quad (6)$$

Where, Q (S, A) is the Q-value at state S for action A, and for computing Q- value immediate reward R(S, A) and maximum Q-value from the next state S' required. gamma (γ) is a discount factor that decides the importance of future rewards [7, 10]. The value of Q (S', A) depends upon future
Q-values, as defined in equation (7)

$$Q(S,A) = \gamma Q(S',A) + \gamma^2 Q(S'',A) \ldots \gamma^n Q(S''\ldots n, A) \quad (7)$$

For computing Q-value for action A_t at state S_t value of maximum Q action $argmax_{A'}$. $Q(S',A')$ for state S' is required followed by the concepts of exploitation. To update
the value of Q-value equation (8) is defined

$$Q(S_t,A_t) = Q(S_t,A_t) +$$

$$Q(S_t,A_t) = Q(S_t,A_t) + \alpha * [R_{t+1} + \gamma * \max_{A_{t+1}} Q(S_{t+1},A_{t+1}) - Q(S_t,A_t)] \quad (8)$$

In reinforcement learning, the agent must choose whether to continue with the current knowledge about the state, actions, and rewards or to explore other options. Exploration is a greedy approach where the agent focuses on improving their knowledge about the environment for long-term benefits. While in exploitation, the agent tries to compute maximum rewards by exploiting current knowledge rather than knowledge gathering. So, in exploration, the agent persistently gathers information to obtain optimal results while in exploitation, the agent optimizes the decision based on current information available.

V. Rewards

In reinforcement learning, reward refers to feedback or numerical value that is generated by the environment and received by the agent after acting on a particular state. The reward function helps the agent to learn about the environment and update its knowledge about the system. The primary goal of the agent is to choose the action state pairs in a way that leads toward the maximum or minimum reward. In this paper, for the performance measures of Q learning for the cart pole problem, Mean Squared Error (MSE) and Mean Absolute Error (MAE) are applied to guide the RL agent for optimal decision-making policy.

- Mean Squared Error Loss (MSE):

  Mean Squared Error measures the average value of the squared difference between the predicted and the actual value. Mathematical formulation for computation of mean square error is defined in equation (9)
  MSE

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (9)$$

Where, y_i and $\hat{y}_i$ represents the predicted value and actual value of the sample and N is the total no. of samples.

- Mean Absolute Error Loss (MAE)

  MAE evaluates the average of absolute difference between observation entities to the prediction entities. The formula is defined in equation (10):

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \quad (10)$$

Where, N is the total number of samples and y_i and $\hat{y}_i$ are the predicted value and actual value of the particular sample.

VI. Experiments and Results

This section provides the details of experiments and their outcomes after performing various reward functions described in previous sections.

A. Hyperparameters Setting

The results of both reward functions MSE and MAE for cart pole problem solved by the Q learning approach are presented in this section. The validation of the training process for both reward functions is performed by varying hyperparameters and their adjustments for better convergence. The details of hyperparameters are stated in Table I.

TABLE I. The Q learning algorithms Parameter details for cart pole OpenAI environment

Parameters	Value
Gamma	0.99
Episodes	100
epsilon	0.99
Activation function	Tanh, linear
Learning rate	1e-2

B. Performance of various reward functions in Q learning

To evaluate the performance of the Cartpole environment, two reward functions Mean Square Error and Mean Absolute error implemented as reward functions in Q learning algorithms. In the training procedure 100 episodes were generated and based on that samples mean, and median were computed for reward functions as shown in Table II

TABLE II. THE CART POLE'S REWARD FUNCTIONS

Reward Functions	Mean	Max	Min	Median
MSE	26.61386	88	9	22
MAE	25.36634	85	8	20

Figure 4 and Figure 5 show the reward functions MSE and MAE for Q learning algorithms applied to the environment of the Cart pole. The violin plot is a combination of a box plot and a probability density function. The white dot in the box in Figure 4 depicts the median for a specific reward function and the distribution of the reward function is defined by violin plots. The boxplots are used for uniform distribution while the violin plot reveals their different distribution. The violin plot shown in Figure 4 proves that Mean Square Error gave weightage to each outlier value while MAE ignored them. The shape of violin plots shows that for MSE and MAE reward is distributed near the mean value.[9]

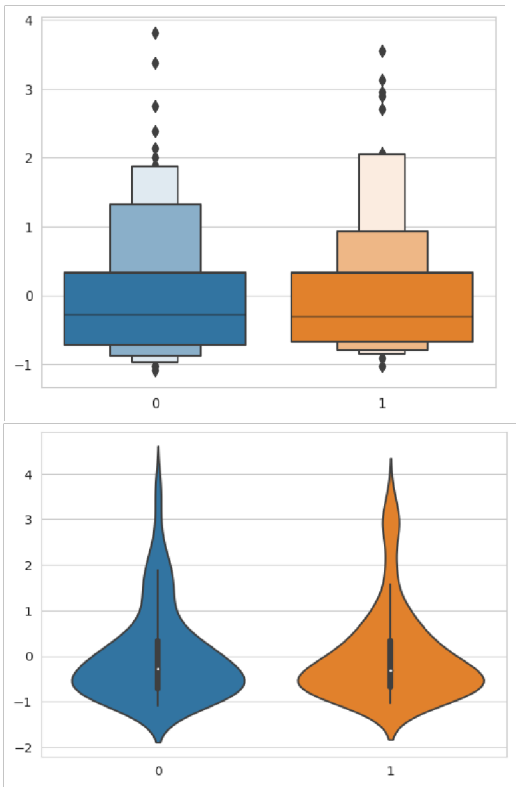

Fig. 4. MSE Q-learning plots(0 = MSE Reward, 1= MAE Reward)

In the figure 5(a), the performance of MSE is depicted. Initially, up to 20 episodes value of the reward function varies between 35 and 7 and the maximum reward for MSE is 88 at episode 72. Figure 5(b) shows the reward and episode plots for MAE reward functions. This graph shows the maximum reward is 85 but MAE ignores the outlier values.

Figure 5(c) is a comparative line plot of both reward functions MSE and MAE. Table II shows the comparative details about both MSE and MAE reward functions which include mean, median, max, and min for both MSE and MAE.

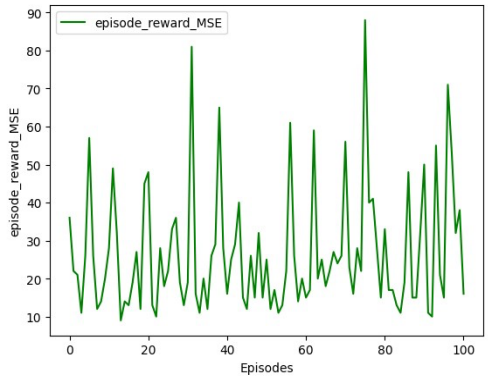

Figure 5(a): MSE Q learning performance

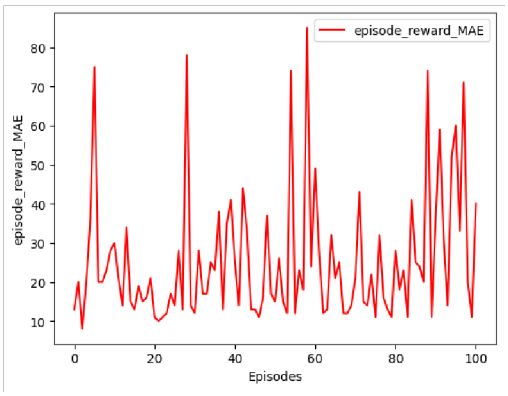

Figure 5(b): MAE Q learning performance

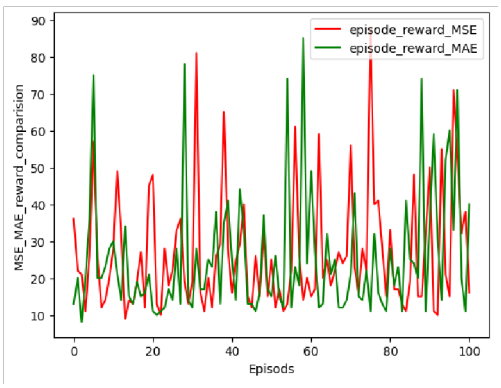

Figure 5(c): Comparative plot for MSE Q learning and MAE Q Learning performance

VII. CONCLUSION

Reinforcement Learning approaches implement the mathematical methodology for computing optimal solutions to determine the best decision-making strategies for agents. The agent is trained for future perspective according to those strategies and seeks the best solution in a specific scenario. In this paper, Q-learning with Mean Squared Error (MSE) and Mean Absolute Error (MAE) as reward functions are applied to the cart pole system. The performance evaluation of both proposed approaches is based on balancing of pole with maximum reward. According to that Q-Learning with MAE ignores the outliner values while Q-Learning with MSE gives importance to outlier values. In future work, more RL models can be applied to the Cart Pole problem and compare their performance.

REFERENCES

[1] Mishra, S., & Arora, A. (2023). A Huber reward function-driven deep reinforcement learning solution for cart-pole balancing problem. Neural Computing and Applications, 35(23), 16705-16722.

[2] Mishra, S., & Arora, A. (2022). Double Deep Q Network with Huber Reward Function for Cart-Pole Balancing Problem. International Journal of Performability Engineering, 18(9), 644.

[3] Kumar, S.: Balancing a cartpole system with reinforcement learning–a tutorial. arXiv preprint arXiv:2006.04938 (2020)

[4] Gym, O., Sanghi, N.: Deep reinforcement learning with python.

[5] Samsuden, M. A., Diah, N. M., & Rahman, N. A. (2019, October). A review paper on implementing reinforcement learning technique in optimising games performance. In 2019 IEEE 9th international conference on system engineering and technology (ICSET) (pp. 258263). IEEE.

[6] Jia, J., & Wang, W. (2020, October). Review of reinforcement learning research. In 2020 35th Youth Academic Annual Conference of Chinese Association of Automation (YAC) (pp. 186-191). IEEE.

[7] Shi, Q., Lam, H. K., Xiao, B., & Tsai, S. H. (2018). Adaptive PID controller based on Q‐learning algorithm. CAAI Transactions on Intelligence Technology, 3(4), 235-244.

[8] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). Openai gym. arXiv preprint arXiv:1606.01540.

[9] Ada, S. E., & Ugur, E. (2023). Meta-World Conditional Neural Processes. arXiv preprint arXiv:2302.10320.

[10] Nagendra, S., Podila, N., Ugarakhod, R., & George, K. (2017, September). Comparison of reinforcement learning algorithms applied to the cart-pole problem. In 2017 international conference on advances in computing, communications and informatics (ICACCI) (pp. 26-32). IEEE.

[11] Ladosz, P., Weng, L., Kim, M., & Oh, H. (2022). Exploration in deep reinforcement learning: A survey. Information Fusion, 85, 1-22.

[12] Huang, X. (2022). Opponent cart-pole dynamics for reinforcement learning of competing agents. Acta Mechanica Sinica, 38(5), 521540.

[13] Mothanna, Y., & Hewahi, N. (2022, November). Review on Reinforcement Learning in CartPole Game. In 2022 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT) (pp. 344-349). IEEE.