EPiC Series in Computing
Volume 98, 2024, Pages 36–45
Proceedings of 39th International Conference on Computers and Their Applications

Rear-end Vehicle Collision Avoidance using Reinforcement Learning

Nashwan Sebi, Narendra Kintali, and Ka C Cheok

Oakland University, Michigan, U.S.A.
njsebi@oakland.edu, kintali@oakland.edu, cheok@oakland.edu

Abstract

A rear-end collision happens when a driver collides with the vehicle directly ahead of them from the rear. Such accidents are common at traffic stops like red lights and stop signs or in heavy traffic conditions. While most rear-end accidents occur at low speeds, they can also happen at higher speeds on highways, interstates, and similar fast-moving roadways. Typically, these accidents involve two vehicles, but they can sometimes lead to a domino effect involving multiple vehicles. [1]. This study delves into Mitigating rear-end vehicle collisions using reinforcement learning (RL). The RL algorithm in focus is intended to be integrated into the ego vehicle's system, see Figure 1, aiming primarily to avert colliding with the rear car when both vehicles are progressing forward. Through the utilization of reinforcement learning algorithms, the RCA system can learn from its interactions with the environment, adapt to changing scenarios, and make intelligent decisions to prevent or mitigate collisions effectively. This research investigates the application of the Deep Deterministic Policy Gradient (DDPG) algorithm in the context of rear collision avoidance. The research methodology involves developing a simulated environment that accurately represents lane driving scenarios using longitudinal car dynamics for the ego and rear cars, including the two vehicles' speeds and positions. The outcomes of this research study are expected to contribute to the development of advanced rear collision avoidance systems that can adapt and improve their performance based on real-time data and experiences.

Keywords: Rear Collision Avoidance (RCA); Reinforcement Learning (RL); Deep Deterministic Policy Gradient (DDPG); Deterministic Policy Gradient (DPG); Rear-end Collision Avoidance (RCA); Deep Q-Network (DQN); Automatic Emergency Braking (AEB). Rear Automatic Emergency Braking (AEB-rear).

1 Introduction

Rear-end collision avoidance is incredibly significant when it comes to ensuring road safety. It refers to the technology and systems designed to help prevent or mitigate collisions from occurring from behind a vehicle. Here are a few key reasons why rear collision avoidance is so important:

1. Preventing accidents: Rear-end collisions are one of the most common types of accidents on the road. Having rear collision avoidance systems in place can significantly reduce the likelihood of such accidents. This technology can detect objects or vehicles approaching from behind and provide warnings or take automatic actions to prevent a collision. Around 1.7 million rear-end collisions, about 29 percent of all car crashes, occur in the United States annually [1].

2. Mitigating deaths and injuries: Rear-end collisions can result in whiplash, head, neck, and back injuries, which can have long-lasting effects on the individuals involved. Rear collision avoidance systems can help by triggering automatic braking or alerting the driver to act, reducing the severity of the impact and potentially minimizing deaths and injuries; around 1,700 people die, and another 500,000 are injured in rear-end collisions in the United States each year [1].

3. Protecting vulnerable road users: Rear collision avoidance systems detect other vehicles and can also recognize cyclists or motorcyclists approaching from behind. By providing timely warnings or interventions, these systems can help protect vulnerable road users who may not have the same protection as vehicle occupants.

4. Enhancing driver convenience: Having an active Rear collision avoidance system helps the drivers focus on other tasks by mitigating worrying about collisions by a vehicle from the rear end.

5. Reducing insurance costs: With rear collision avoidance systems becoming more common, insurance companies often offer reduced premiums for vehicles equipped with such safety features. This can provide financial incentives for drivers to prioritize vehicles with rear collision avoidance technology, promoting its adoption and overall road safety.

Overall, rear collision avoidance is vital in preventing accidents, reducing injuries, protecting vulnerable road users, enhancing driver convenience, and promoting safer driving practices. Utilizing this technology can create a safer and more secure road environment for everyone.

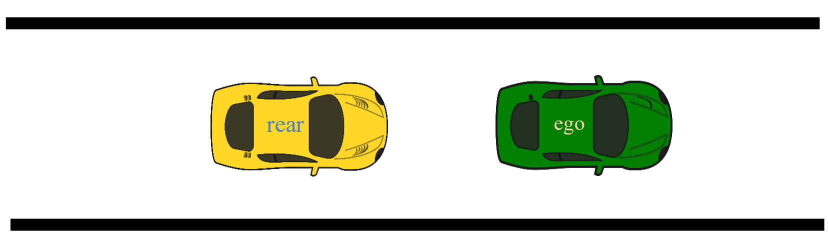

Figure 1: Ego and rear cars setup

2 Problem statement

Background: Rear-end collisions remain among the most predominant types of vehicular accidents across urban and highway settings. Using some systems has alleviated the issue; for example, traditional techniques like (AEB-rear) are used in ego cars that are automatically applied to prevent a rear-end collision while the car travels in reverse but not forward. Some other traditional systems are implemented in the rear cars, like AEB and forward collision warnings. However, the conventional deterministic solutions can sometimes falter when faced with unpredictable driving patterns, evolving traffic conditions, and a diverse range of vehicle dynamics.

Problem: Current systems are predominantly installed in the rear cars and aim to help avoid colliding with the ego cars, and these systems operate under rule-based or deterministic frameworks, relying heavily on pre-defined conditions to trigger warnings or corrective actions. While effective in many scenarios, these systems might be inadequate in dealing with situations that haven't been explicitly programmed for. They might also struggle in scenarios with multi-agent interactions where the behavior of other vehicles (agents) isn't strictly predictable. This limitation hints at the need for intelligent systems installed in the ego car that can learn and adapt in real-time, making decisions based on a continually evolving understanding of traffic dynamics. Integrating these intelligent systems in the ego (leading) cars and having rule-based systems in the rear cars will contribute to a safer driving experience with all the additional benefits that could be earned from this integration.

Reinforcement Learning (RL) as a Solution: RL is a branch of machine learning that operates on the principle of learning optimal strategies through trial and error by interacting with an environment. For rear-end collision avoidance, an RL agent could continually adapt its decisions based on past experiences, making it well-suited to address the unpredictable nature of real-world driving. By training in a simulated environment that mimics real-world conditions, RL agents could learn the nuances of driving behaviors and develop safe and efficient strategies.

3 Approach

3.1 Software and Hardware used in the study

In this study, we utilized MATLAB R2023b and Simulink alongside a suite of relevant MATLAB toolboxes, such as the Reinforcement Learning Toolbox. To handle the computationally demanding tasks, we employed computers outfitted with multi-core GPUs, which are adept at swiftly managing and modifying memory to boost processing speed. Furthermore, to optimize time efficiency, these two machines, Dell (Alienware m17 R4) and MSI (GE65 Raider 95F), were deployed to execute various training algorithms concurrently. See Figure 2.

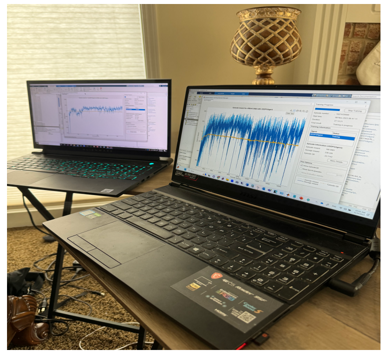

Figure 2: Computers used in the research.

3.2 Reinforcement Learning (RL)?

Reinforcement learning aims to teach an agent how to successfully accomplish a task in an environment where certainty is not guaranteed. In this process, the agent continuously receives feedback from the environment in the form of observations and rewards at every time step. These rewards serve as instantaneous evaluations of the agent's previous action based on its effectiveness in progressing toward the task's objective. The agent responds by adjusting its actions according to the information received [2]. The interaction between the agent and the environment occurs at distinct intervals, as shown in the sequence of discrete time steps depicted in Figure 3.

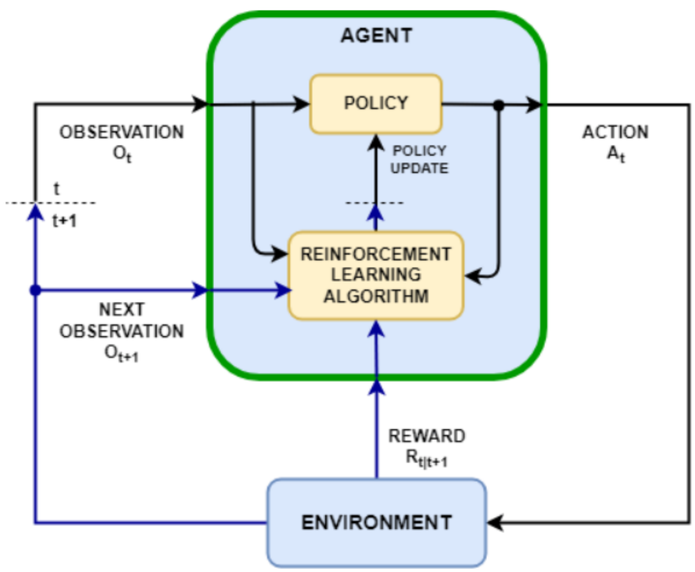

Figure 3: Reinforcement Learning Agent and the environment interaction. Image credits to [3].

3.3 Deep Deterministic Policy Gradient (DDPG)

In our research, we employed the Deep Deterministic Policy Gradient (DDPG) as our choice of reinforcement learning algorithm. DDPG is a synthesis of DPG and DQN principles, making it highly effective in scenarios with continuous action spaces, aligning well with our application's requirements [4]. The fundamental elements of DDPG include:

a) **Deterministic Policy Gradient**: Unlike conventional reinforcement learning algorithms that generate a range of possible actions, DPG provides specific actions for given states. This deterministic approach is more suitable for continuous action environments [5].

b) **Deep Learning Integration**: DDPG utilizes deep neural networks to manage complex, high-dimensional state and action spaces [6]. These networks are trained to approximate both the policy (decision-making process) and the value (reward evaluation) functions, harnessing the power of deep learning [7].

c) **Actor-Critic Architecture**: The algorithm adopts an actor-critic framework. The 'actor' is trained to choose the best possible action in a given state, while the 'critic' assesses the value of the action-state pair. This bifurcation helps in effectively balancing the exploration of the environment with the exploitation of acquired knowledge [8].

d) **Experience Replay**: DDPG uses a replay buffer to store past experiences, comprising states, actions, rewards, and subsequent states. Learning from randomly sampled experiences from this buffer reduces correlations between successive learning instances, enhancing stability and learning efficiency [9].

e) **Target Networks**: The algorithm employs distinct 'target' networks for both the actor and the critic. These networks are updated incrementally, which aids in stabilizing the training process by ensuring gradual changes in learning targets.

DDPG has proven to be particularly beneficial in our application, which demands accurate and continuous control, showcasing its adaptability and effectiveness in complex environments.

3.4 Goal of the Intelligent Rear-end Collision Avoidance (RCA)

The training aims to teach the ego car to consistently maintain a predetermined (set) speed and avoid collisions by staying at a safe distance from the rear car, see Figures 4 and 5 below. This is achieved through various maneuvers such as acceleration, deceleration, and braking.

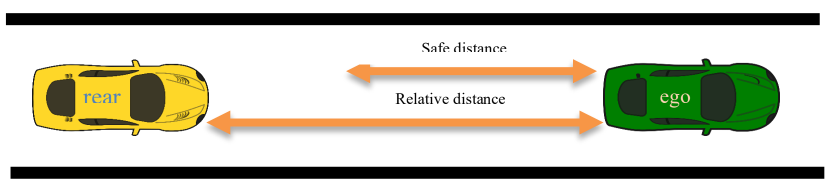

Figure 4: Speed control, the training objective is the speed of ego = determined speed.

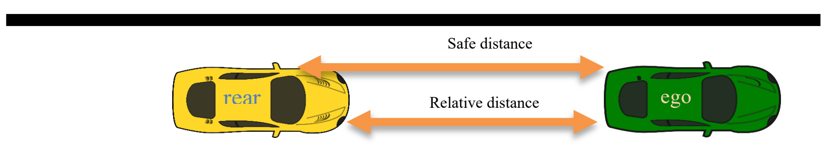

Figure 5: Spacing control, the training objective is the relative distance = safe distance.

3.5 Simulink Model of the two vehicles

This research employs the identical vehicle model that is used in the example of the Adaptive Cruise Control System using Model Predictive Control from the Model Predictive Control Toolbox, as referenced in [12].

3.6 Steps for Train DDPG Agent for RCA

1) Define the starting position and initial speed for both vehicles.
2) Define the default spacing at a standstill (in meters), the time gap (in seconds), and the determined velocity by the ego driver (in meters per second).
3) Determine the sample time, T_s and the total duration of the simulation, T_f, both expressed in seconds.
4) Establish the acceleration limit, considering the physical constraints of the vehicle model being utilized.
5) Formulate the specifications for the observation and action.
6) Develop the interface for the environment.
7) To randomize the rear car's initial position, a function was created for this purpose.
8) Set a specific seed for the random number generator to ensure reproducibility.
9) Develop a DDPG agent.
10) Train the agent.
11) Validate the performance of the trained agent.

3.7 RCA strategy to maintain a safe distance from the rear car

When the relative distance between the ego car and the rear car falls below the safe distance threshold, the ego car adjusts its speed to match whichever is higher: the velocity of the rear car or the velocity set (determined) by the ego vehicle driver. This approach helps the ego car to maintain a certain distance from the rear car. Conversely, if the relative distance exceeds the safe distance, the ego car aligns its speed to the velocity predefined by the driver.

3.8 RCA Agent Training

After training multiple agents, we successfully reached the optimal agent; the training progress of the optimal agent is illustrated in Figure 6. This achievement involved fine-tuning several parameters, including the number of episodes (representing the entire sequence of an agent's actions and interactions in the environment), stopping criteria for training, settings for preserving the agent's value, and adjusting the learning rates for both the actor and critic. Furthermore, we modified the noise's variance and decay rate. Additionally, we introduced weight decay for both the actor and critic networks, a technique designed to mitigate overfitting by penalizing larger weights within the model.

3.9 Training Progress

Monitoring the training progress of the RCA was beneficial. Visualizing key metrics during training through plots, as shown in Figure 6, provided insights into the ongoing training dynamics. In RCA's RL training, the agent started with random actions and gradually refined its policy by learning from the consequences. It adapted its behavior over iterations, aiming to maximize cumulative rewards and becoming increasingly proficient at making optimal decisions.

Here's a concise explanation of the three colors in the RCA training progress graph:

Light Blue: Represents the actual performance of the RL agent during training, indicating how its capabilities improve over time. You can see it's relatively consistent and keeps high reward levels of the actual performance over 20000 episodes.

Dark Blue: Signifies a smoothed or averaged version of the light blue curve, providing a clearer trend of the agent's progress by reducing noise, and you can notice the same thing here: you can see the consistency and high average levels over the same number of episodes.

Orange: Serves as a reference or baseline, representing a predefined or benchmark performance level for comparison against the agent's learning achievements.

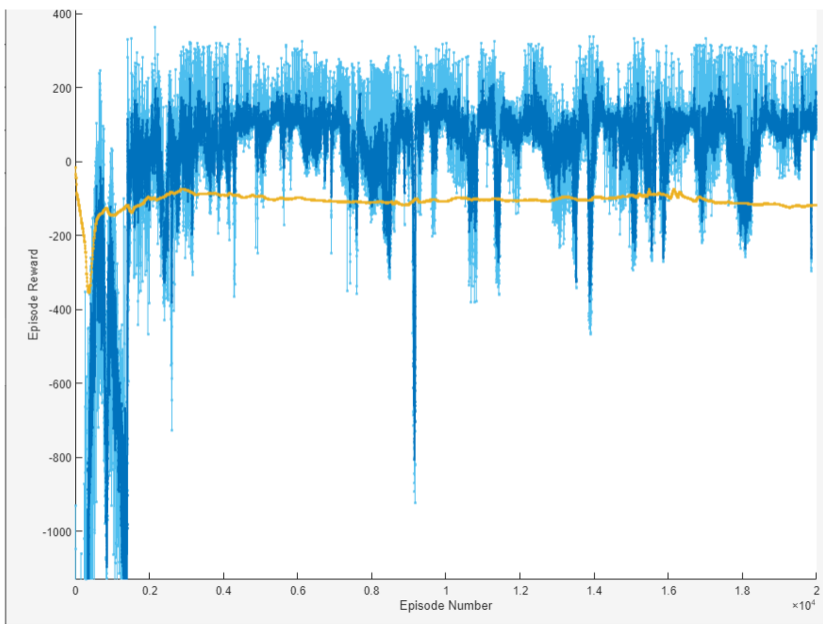

Figure 6: Training process of the optimal agent obtained in this study for the RCA

3.10 Saving the trained agent of the RCA

Another MATLAB code was created to save the trained agents.

3.11 Validating the performance of the trained agent

We conducted simulations within the Simulink environment to assess the trained agent's performance.

4 Results

The provided plots, Figure 7 and Figure 8 illustrate simulation results when the rear car is 80 meters behind the ego car.

During the initial approximately 47 seconds, the relative distance exceeds the safe threshold (upper plot), prompting the ego car to match its predetermined velocity (middle plot). Between roughly 47 and 58 seconds, the relative distance falls below the safe distance (upper plot), leading the ego car to follow the higher value between the rear car's velocity and the predetermined velocity. Consequently, at roughly second 47, the ego car exhibits a sudden acceleration to catch up with the rear car's speed (bottom plot) and continues to mirror the rear car's speed as long as the condition holds. Subsequently, the ego car adjusts its speed to either track the higher of the rear car's velocity and the predetermined velocity or simply the predetermined velocity based on the perceived safety of the relative distance.

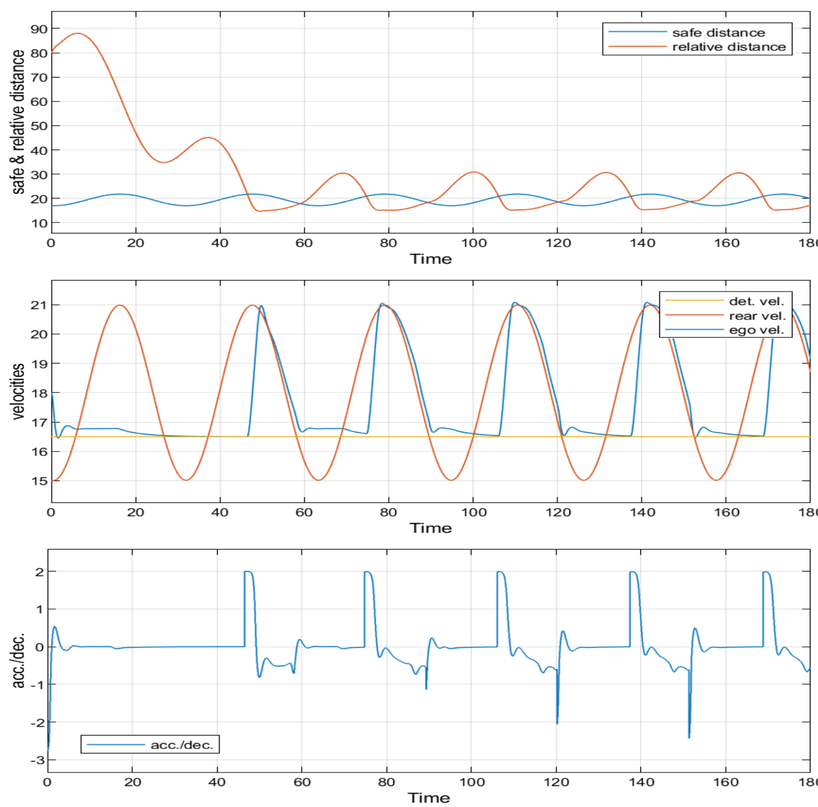

Figure 7: RCA Simulation results when the rear 80 meters behind the ego car

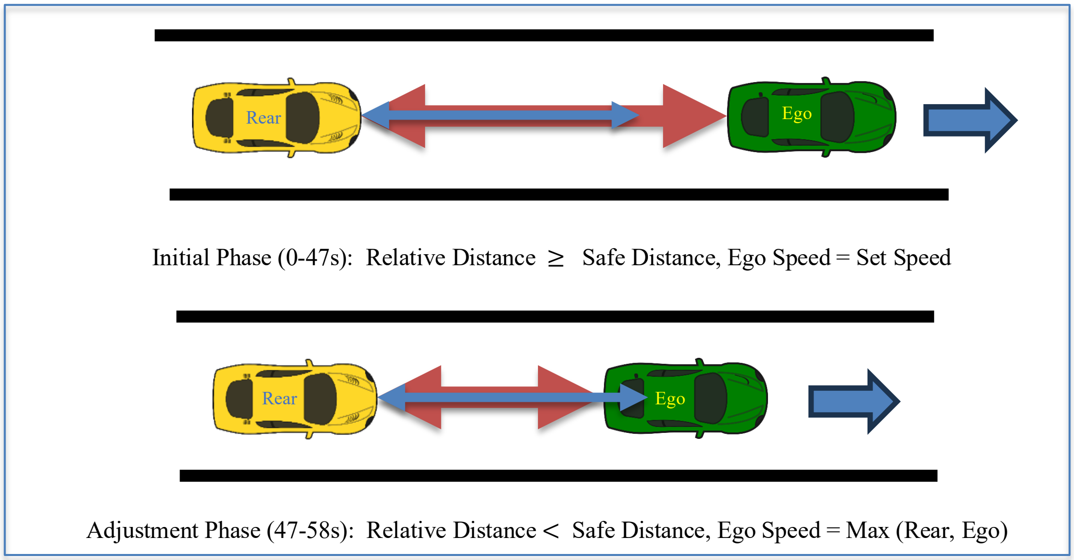

Figure 8: Ego Vehicle Speed Adjustment of RCA System

5 Conclusion and next phase

This study aimed to ascertain the applicability of Reinforcement Learning (RL) within an Intelligent Rear-end Collision Avoidance (RCA) system integrated into an ego car aimed at mitigating rear-end collisions. We developed the necessary code and utilized pre-existing Simulink models for both the ego and rear cars to train an RL agent based on a specific strategy tailored for this purpose. Subsequently, we assessed the trained agent's performance by resetting various parameters, including positions, speeds, and spacing distances. The initial findings are very promising. Our next phase entails incorporating a lead car positioned in front of the ego vehicle and enhancing the algorithm to create an Intelligent Forward and rear Collision Avoidance (I-F&RCA) system as depicted in Figure 9.

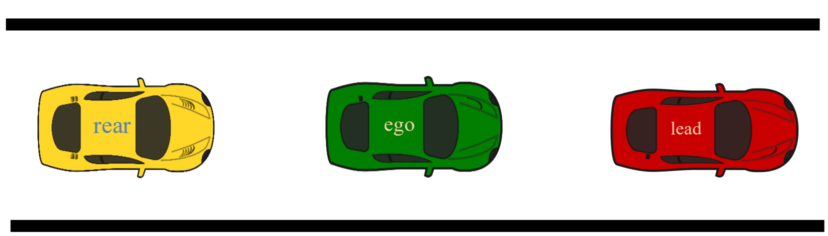

Figure 9: Intelligent Forward and rear Collision Avoidance (I-F&RCA)

References

[1] David Abels, Abels and Annes, PC Personal injury lawyers. [online]. Available: https://www.daveabels.com/rear-end-collisions-frequent-type-collision/#:~:text=In%20fact%2C%20roughly%201.7%20million,are%20injured%20in%20the%20crashes.

[2] Sutton, Richard S., and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Second edition. Adaptive Computation and Machine Learning. Cambridge, Mass: The MIT Press, 2018. [online]. Available: https://www.andrew.cmu.edu/course/10-703/textbook/BartoSutton.pdf

[3] MathWorks, Reinforcement Learning Environments [online]. Available: https://www.mathworks.com/help/reinforcement-learning/ug/reinforcement-learning-environments.html

[4] W. Chao, D. Han and X. Jie, "Multi-Rotor UAV Autonomous Tracking and Obstacle Avoidance Based on Improved DDPG," 2021 2nd International Conference on Artificial Intelligence and Computer Engineering (ICAICE), Hangzhou, China, 2021, pp. 261-267, doi: 10.1109/ICAICE54393.2021.00059.

[5] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, Martin Riedmiller in Proceeedings of the 31st International Conference on Machine Learning, PMLR 32(1):387-395, 2014.

[6] V. Sze, Y. -H. Chen, T. -J. Yang and J. S. Emer, "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," in Proceedings of the IEEE, vol. 105, no. 12, pp. 2295-2329, Dec. 2017, doi: 10.1109/JPROC.2017.2761740.

[7] P. V. R. Ferreira et al., "Multiobjective Reinforcement Learning for Cognitive Satellite Communications Using Deep Neural Network Ensembles," in IEEE Journal on Selected Areas in Communications, vol. 36, no. 5, pp. 1030-1041, May 2018, doi: 10.1109/JSAC.2018.2832820.

[8] A. Dolatabadi, H. Abdeltawab and Y. A. -R. I. Mohamed, "An Improved Actor-Critic Reinforcement Learning with Neural Architecture Search for the Optimal Control Strategy of a Multi-Carrier Energy System," 2023 IEEE PES GTD International Conference and Exposition (GTD), Istanbul, Turkiye, 2023, pp. 315-319, doi: 10.1109/GTD49768.2023.00086.

[9] S. M. Seyed Motehayeri, V. Baghi, E. M. Miandoab and A. Moeini, "Duplicated Replay Buffer for Asynchronous Deep Deterministic Policy Gradient," 2021 26th International Computer Conference, Computer Society of Iran (CSICC), Tehran, Iran, 2021, pp. 1-6, doi: 10.1109/CSICC52343.2021.9420550.

[10] MathWorks, Deep Deterministic Policy Gradient (DDPG) Agents. [online]. Available: https://www.mathworks.com/help/reinforcement-learning/ug/ddpg-agents.html.

[11] Lillicrap, Timothy P., Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. “Continuous Control with Deep Reinforcement Learning.” *ArXiv:1509.02971 [Cs, Stat]*, September 9, 2015.

[12] MathWorks, Adaptive Cruise Control System Using Model Predictive Control. [online]. Available: https://www.mathworks.com/help/mpc/ug/adaptive-cruise-control-using-model-predictive-controller.html.