SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Fluid Mechanics and Thermal Sciences

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Comparison Method Q-Learning and SARSA for Simulation of Drone Controller using Reinforcement Learning

Mohamad Hafiz Abu Bakar[1], Abu Ubaidah Shamsudin[1,*], Ruzairi Abdul Rahim[1], Zubair Adil Soomro[1], Andi Adrianshah[2]

[1] Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, Malaysia
[2] Universitas Mercu Buana Jakarta, Jl. Raya, RT.4/RW.1, Meruya Sel., Kec. Kembangan, Jakarta, Daerah Khusus Ibukota Jakarta, 11650, Indonesia

ARTICLE INFO

Article history:
Received 20 November 2022
Received in revised form 9 April 2023
Accepted 17 April 2023
Available online 6 May 2023

Keywords:
Reinforcement Learning (RL); Q-Learning; SARSA(State-Action-Reward-State-Action); drone

ABSTRACT

Nowadays, the advancement of drones is also factored in the development of a world surrounded by technologies. One of the aspects emphasized here is the difficulty of controlling the drone, and the system developed is still under full control by the users as well. Reinforcement Learning is used to enable the system to operate automatically, thus drone will learn the next movement based on the interaction between the agent and the environment. Through this study, Q-Learning and State-Action-Reward-State-Action (SARSA) are used in this study and the comparison of results involving both the performance and effectiveness of the system based on the simulation of both methods can be seen through the analysis. A comparison of both Q-learning and State-Action-Reward-State-Action (SARSA) based systems in autonomous drone application was performed for evaluation in this study. According to this simulation process is shows that Q-Learning is a better performance and effective to train the system to achieve desire compared with SARSA algorithm for drone controller.

1. Introduction

Nowadays, there are various studies carried out by previous researchers involving autonomous drones and using different technologies that have been introduced to the drone applications to enable it for the flying process without full user control [1,2]. Most of the studies conducted indicate the trends identified by some researchers in the new technologies that have been implemented to improve drone abilities using AI technology. Machine Learning is one of the effective branches in AI, which is practically can be used for training the robot without complete human supervision [3]. To enable the system to operate automatically, Reinforcement Learning enables the agent (drone) to learn the next action based on the interaction between the agent and the environment.

Through the development of technology based on Reinforcement Learning, the system developed can operate fully automatically and more effectively. There are various algorithms in the Reinforcement Learning branch, including Q-Learning, SARSA, DQN (Deep Q-Learning), and DDPG

* *Corresponding author.*
E-mail address: ubaidah@uthm.edu.my

https://doi.org/10.37934/araset.30.3.6978

(Deep Deterministic Policy Gradient) [4-8]. In this paper, we are exploring the capabilities or potential that can develop in an autonomous drone. Due to the high similarity of the algorithm structures, Q-Learning and SARSA are used in this study and the comparison of results involving both the performance and effectiveness of the system based on the simulation of both methods can be seen through the analysis.

The objective of this study is to develop two systems based on Q-learning and SARSA for an autonomous drone application. In addition, comparing both of the methods will robustly contribute to the development of drone especially in term of the exploration of knowledge related to Reinforcement Learning.

This paper is an extension of the study originally presented by Bakar *et al.,* [9] that previously highlighted the optimization hyperparameter for the system developed based on the Q-learning algorithm. The advancement of drones is also factored in the development of a world surrounded by technologies. One of the advantages of drone applications to be considered is the variety of uses of the drone itself including firefighter drone application [10]. But there are also limitations which can give an opportunity to improve or renew in the evolution of technology, which can further increase the potential of the drone itself. One of the aspects that are emphasized here is related to the difficulty of controlling the drone and as well the systems developed are still under full controller by the users.

2. Related Works

This section discusses more about previous works or research that is related to this paper in Reinforcement Learning particularly Q-Learning and the SARSA algorithm. In addition, drone applications that are based on machine learning algorithms for their controlling systems, are briefly discussed.

In this discussion, the attention is on Reinforcement Learning, in which one of the Machine Learning systems provides a framework that is able to learn automatically by interacting with the environment based on previous experience. In this article by Che-Cheng *et al.,* [2] from 2019, they use the ArUCO markers method as a reference for improving the accuracy of drones in the take-off, fly forward and landing processes by applying Q-learning as the main approach of their system. Simulations using the ROS and Gazebo software platforms demonstrate that the proposed method has been studied to improve the efficiency and accuracy of the system.

Therefore, from the work presented by Meerza *et al.,* [4] suggested combining the PSO (Particle Swarm Optimization) technique with Q-Learning for improving the performance of the system developed for mobile robots. The outcomes in this paper study highlight the use of Q-Learning and display significant differences in combination with the previous method where the Q -Learning has successfully improved the learning process and system accuracy in the robot for the unknown environment.

Furthermore, Karthik *et al.,* [11] an RL approach was proposed to overcome the stability issue of the quadrotor. The mechanism starts with the drone learning to hold the specified altitude using Q-Learning and then using Q-Learning and PID for the collision avoidance process. The purpose of this technique is to make a valid systematic comparison between the Q-Learning and PID controlling. The results indicate that Q-Learning is better than PID and provides robust performance to avoid obstacles in an environment unknown to the system.

Cheng *et al.,* [12] expressed that the SARSA algorithm can avoid collisions from an enemy UAV during the flying phase to the target point. The learning process framework was developed to train UAV to avoid collision accidents by using Temporal Difference based on the SARSA method. The

results of the simulation indicate that the UAV agent is able to learn the appropriate policy under the proposed system.

Although the comparison between the SARSA technique and the Temporal Difference (TD) in a robotic system to avoid dangerous situations was discussed by Harwin and Supriya [13] in article, in this comparison, SARSA has a better success rate than TD after the evaluation of simulation results using MATLAB for both methods. In addition, the study shows that the SARSA successfully solves complex tasks in order to avoid obstacles, particularly dangerous situations.

A paper in 2019 published by Sichkar [14] conducted a study for two types of RL methods, Q-Learning and SARSA, for a mobile robot. Sichkar [14] focused on the highest payoff selection in order to accomplish the target with the tasks to avoid obstacles. Based on the result of the simulation, Q-Learning is better for solving the task with faster processing time, while SARSA shows better accuracy to make a decision.

Moreover, this paper has introduced OpenAI gym for robotic systems using Q-Learning and SARSA, according to this analysis in article by Zamora *et al.,* [15]. The software used to run the simulation includes ROS and Gazebo simulator. The paper shows various robots developed by implementing Q-Learning and SARSA algorithms in robot systems such as Turtlebot, Erle-Copter and Erle-Rover to prove RL is suitable for different robot structures. Through this paper, it is shown that Q-learning is fast in the learning process, especially in a random environment.

In addition, the comparison between Q-Learning and SARSA in robot applications can be demonstrated by previous work. Furthermore, based on our observations from previous works, comparisons of Q-Learning and SARSA algorithms for drone applications have not been conducted before. In order to determine its effectiveness and performance, particularly in the learning process using the Reinforcement Learning method especially for UAV applications through this research, two algorithms for drone controlling systems will be compared between Q-Learning and SARSA, respectively.

3. Theory

3.1 Introduction

This topic discusses related fundamentals and practically that will be implemented in this project. To control the drone and achieve our objective, it will be implementing Reinforcement Learning (RL) for 2 different methods, which are Q-Learning and SARSA. Theoretically, Temporal Difference (TD) is one of the methods in RL: the model-free method. Basically, TD is an agent that learns through extraction from its environment. On-Policy and Off-Policy are 2 methods of Temporal Difference. In Figure 1 shows the type of Temporal Difference.

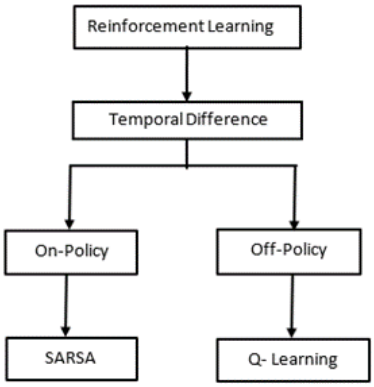

Fig. 1. Block diagram of Temporal Difference (TD)

3.2 Algorithm of Q-Learning & SARSA

The idea is to build a system that can control the drone without a full controller, which is itself operated by the environment learning through the trial-and-error process, depending on the reward and value of the discount based on the tasks performed by the robot. In setting up the system, we will deploy 2 methods of Reinforcement Learning to compare performance and response during the learning process.

In this study, Q-Learning is the first algorithm for Reinforcement Learning to control the system. The Q-Learning algorithm is an Off-Policy algorithm for Temporal Difference Learning, where it is described as follows

$$Q_{new}(S_t, A_t) \leftarrow Q(S_t, A_t) + \{R_{t+1} + \gamma maxQ(S_{t+1}, A_{t+1}) - Q(S_t, A_t)\} \quad (1)$$

Q-Learning learns the optimal policy where the action is chosen based on a more exploratory and random policy. As shown in Eq. (1), the Q-Learning algorithm consists of an agent that interacts with the environment by state. (G_t) is the target for Q-Learning to be updated by Eq. (2) as shown by Sutton and Barto [16] as follows

$$G_t = R_{t+1} + \gamma maxQ(S_{t+1}, A_{t+1}) \quad (2)$$

On the second method of comparison, RL is the SARSA algorithm, where it is an On-Policy algorithm that learns from near-optimal policy compared to Q-Learning directly from the optimal policy. SARSA algorithm as described by Eq. (3).

$$Q_{new}(S_t, A_t) \leftarrow Q(S_t, A_t) + \{R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)\} \quad (3)$$

In this case, the SARSA algorithm for the target update policy (G_t) is similar to the Q-Learning policy, but without considering the maximum reward for the next update as referred to in Eq. (4).

$$G_t = R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) \quad (4)$$

Below is an explanation of how information is related to both of algorithms

where

i. $Q_{new}(S_t, A_t)$ -new Q value for that state and action
ii. $Q(S_t, A_t)$ - current/old Q value
iii. α - learning rate
iv. R_{t+1} - reward for taking that action at that state
v. γ – discount factor of future reward
vi. For Q-learning algorithm, $maxQ(S_{t+1}, A_{t+1})$ - maximum expected future reward given the new state and all possible actions at that new state
vii. For SARSA algorithm, $Q(S_{t+1}, A_{t+1})$ - expected future reward given the new state and all possible actions at that new state

4. Simulation Setup

The simulation method used is very important in interpreting the results at the end of the experiment to achieve the purpose of this project. The framework must be based on the concept of the requirements previously proposed, taking into consideration the implementation structure to be followed. In running the drone simulation using ROS and Gazebo simulator, the structure to run the simulation using OPENAI is described in the diagram in Figure 2.

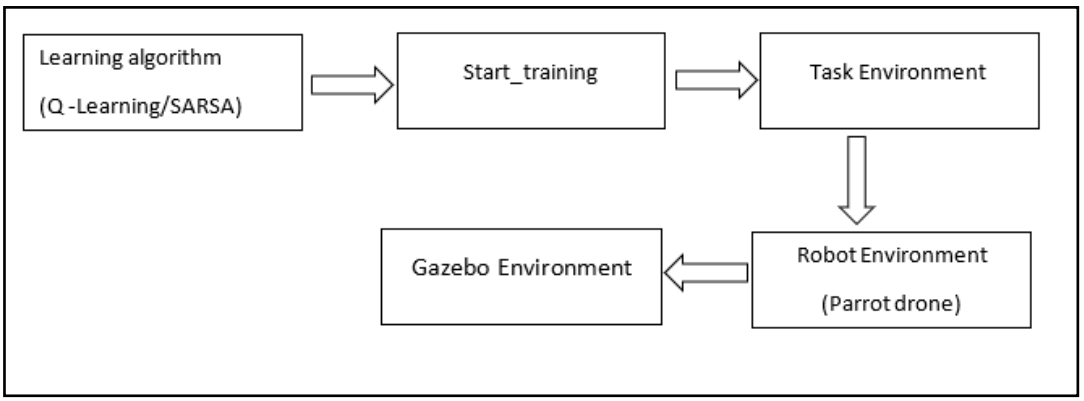

Fig. 2. Block diagram to execute programming using ROS and Gazebo simulator

Accordingly, Figure 2 shows the block diagram for implementing programming using the Reinforcement Learning method. In the session, where the drone is trained to execute instructions, the learning algorithm used in this project must be determined using two different algorithms, the first is Q -Learning and the second is SARSA. After that, we start by developing a training script for the training phase (start-training). To enable drone to perform tasks, training based on structured programming requires a task environment. Then, a robot environment is needed to allow the drone to be used to perform the task. In this case, we use the Parrot drone as an agent to interact with the environment. Thus, the Gazebo environment is used to connect programming to the simulation platform in order to execute all programming instructions and also to communicate with the robot environment in order to provide action instructions for the drone.

One of the most important aspects of running this simulation is the appropriate software required. In order to make this simulation successful, this study uses the ROS and Gazebo simulator to demonstrate that all experiments related to RL can be performed to evaluate the system's effectiveness. In fact, the most important element in this simulation that has been developed is an agent using a Parrot AR.Drone. Most of these drone simulation practices are based on The Construct website where we refer to the implementation of the simulation platform for algorithm concepts and ideas through this website [17]. Table 1 shows information related to the software used in this study.

Table 1
Softwares used in this study

Software	Description	Version
Robotic Operating System (ROS)	A robotic programming platform	ROS Melodic Morenia
Gazebo simulator	Used to running simulation, also for robotic and environment simulation design	9.14.0

For the simulation, input of system is position, *P*= [x, y, z] and output reinforcement learning is velocity, *V*= [V_x, V_y, V_z] as illustrated in Figure 3. *V* will be sent to control the velocity of the drone which has a PID controller based on work in the paper by Zhao and Jiang [18], but the movement of the drone be controlled by using a Reinforcement Learning algorithm.

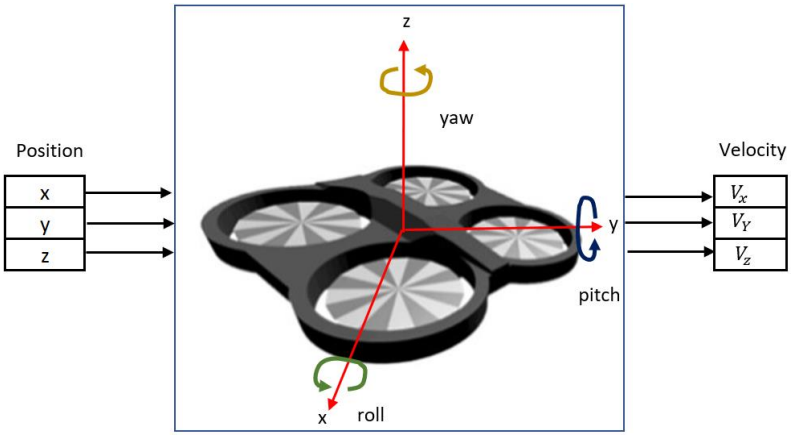

Fig. 3. Input and output the drone for simulation

There are 6 main hyperparameters that determine the learning process. Table 2 reveals all the hyperparameters that our simulation used. The importance of the parameters part for Q-learning and SARSA during the training session. To control drone movements using Q-Learning and SARSA algorithms, the drone must be in direct interaction with the environment. Therefore, a proper simulation system will be setup with the best possible parameters based on the optimization of the hyperparameter suggested in Table 2 [9].

Table 2
Hyperparameter setting for Q-learning and SARSA methods

Parameter	Description	Value
alpha	Learning rate	0.1
gamma	Discount factor of future reward	0.8
epsilon	Exploration constant	0.9
epsilon discount	Value discount for exploration	0.999
nepisodes	Number of episode loop	100
nsteps	Number of step loop	100

Based on this purpose, the hyperparameter will be considered an important part of the first setup to find the best performance during the training session and also can assist agent learn effectively. The reward will be produced by considering the drone movement. In fact, more movement toward the target directly has a higher reward compared to the agent that has moved against the target.

Following Figure 4, shown in the simulation view of a drone, which starts from the initial location, the target location and coordinates based on Table 3. The drone will move randomly through an exploration process from the initial position to the target location by following the policy laid out in the system developed based on interaction with the environment.

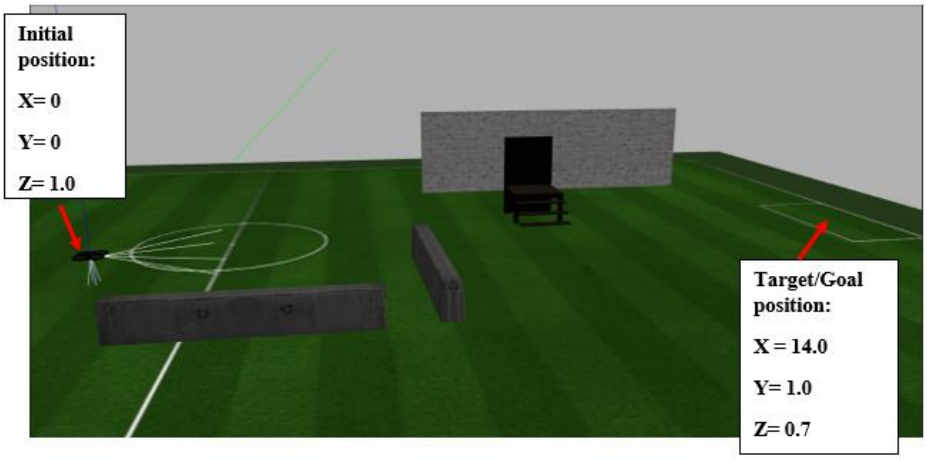

Fig. 4. Initial position and also targeted position

Table 3
Location of initial and target for the drone simulation

Axis	Location (in meter)	
	Initial	Target
X	0	14
Y	0	1
Z	1	0.7

5. Result and Discussion

The results collected from the study will be discussed in this section. The data produced will determine whether we have actually been able to achieve the objective of this study. This chapter also discusses the results of simulation for both algorithms and which one shows better performance in the training process.

Based on Figure 5, a comparison of the overall results for both systems developed with Q-learning and SARSA algorithms is shown. The graph shows that the blue line is a Q-Learning trend that produces better results compared to the red line for SARSA. In fact, most positive reward values are produced by the Q-learning system, which shows that it has a better system within it to generate positive rewards based on the graph produced.

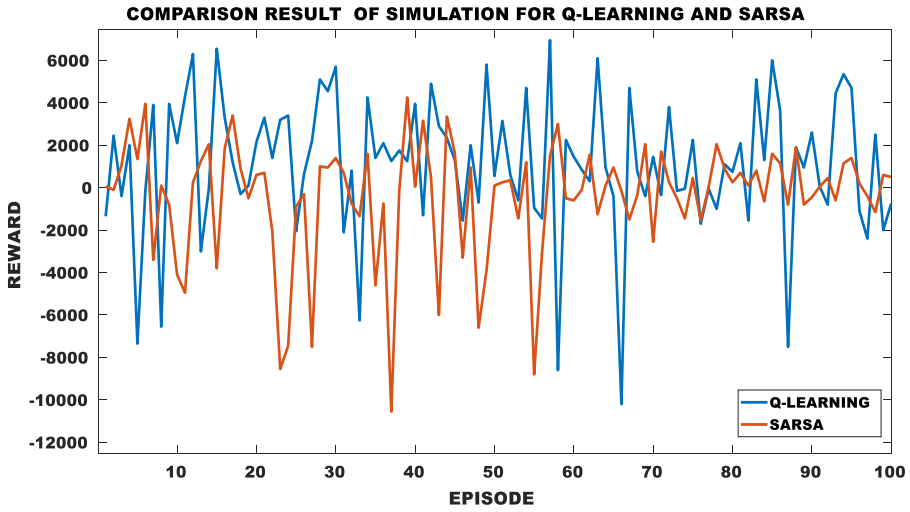

Fig. 5. The result of overall for learning process

The next comparison is based on the highest value generated from both algorithms. Based on Table 4, the highest value for Q-Learning is 6950, while for SARSA it is 4250. While the lowest value for Q-Learning is -10200 and SARSA is -10550, this high reward value reflects the process of moving the drone more positively towards the target compared to the negative value indicating the drone is moving in the opposite direction. Furthermore, Figure 5 shows Q-Learning is better at producing a higher reward value than SARSA.

Table 4
Highest and lowest rewards for both of methods

Reward	Q-Learning	SARSA
Highest Reward	6950	4250
Lowest Reward	-10200	-10550

To see the cumulative value of the simulation executed, Table 5 will describe the total value of the rewards and time taken by the system after completing 100 routine episodes of the simulation. Through the comparison of this cumulative value, one can also differentiate the value of the reward produced overall by the two methods developed through the simulation conducted before. It also describes the cumulative value for a time where the time value for SARSA is lower than Q-Learning due to a less exploratory factor during the learning process, resulting in a decreased reward value. In addition, the evaluation of the system's performance in the context of the cumulative value of time does not consider the effectiveness and success of that system.

Table 5
Cumulative total for reward and time for the learning process

Type of Reinforcement	Cumulative Total of Reward	Cumulative of Time (In Second)
Q-Learning	112000	3470 s
SARSA	-43400	2794 s

Based on the discussion and comparison of the values generated during the simulation, it is concluded that Q-Learning is more able to yield better and more positive results than SARSA. In fact, most of the results discussed also demonstrate that Q-Learning is better and provides a more positive data set for developing a more effective system. For next, the framework to be developed later based on this study is the development of an Autonomous Drone Firefighter where will have a more optimal, an efficient and reliable system for the development based on Reinforcement Learning method.

Furthermore, based on the theory, Q-learning is an Off-Policy. Through simulations conducted before, most of the episodes recorded by Q-Learning are better in the exploration process based on the results produced by this study. The reward produced is also best proven that Q-learning showed that the pattern of Off-Policy learns more freedom in the learning process without strict instructions. For SARSA, it is an On-Policy where the learning process is more strictly focused on the instructions provided. Therefore, it can be shown that the result of a positive reward value for drones using the SARSA method is lower in the exploration process due to the policy factor applied to the system. Regarding Q-Learning, it is basically a learning policy that is more oriented towards trial and error, with the exploitation of the environment contributing to the next action. A suggestion for SARSA, that is stricter in its algorithm strategy will help improve performance. The evidence presented thus far supports the idea that the method based on Q-Learning is better compared with SARSA through this study, especially in the drone system, which offers flexibility in the learning process.

6. Conclusions

In this work, we have presented a comparison method between Q-Learning and SARSA to determine their efficiency and performance based on simulation. According to this simulation process, it shows that Q-Learning has better performance and effective to train the system to achieve desire compared with SARSA algorithm for drone controller. Moreover, this study also investigates reliable systems that can be used, especially for the drone. In the future, we will work to improve the learning process with implement Deep Reinforcement learning method into the autonomous firefighter drone using a Deep Q-Learning algorithm.

Acknowledgement

The financial support received from the Graduate Research Grant Scheme (GPPS) (Vot No: H606) & Contract Grant (Vot No: H381), Research Management Centre Universiti Tun Hussein Onn Malaysia.

References

[1] Vankadari, Madhu Babu, Kaushik Das, Chinmay Shinde, and Swagat Kumar. "A reinforcement learning approach for autonomous control and landing of a quadrotor." In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 676-683. IEEE, 2018. https://doi.org/10.1109/ICUAS.2018.8453468

[2] Che-Cheng, Chang, Jichiang Tsai, Lu Peng-Chen, and Chuan-An Lai. "Accuracy Improvement of Autonomous Straight Take-off, Flying Forward, and Landing of a Drone with Deep Reinforcement Learning." *International Journal of Computational Intelligence Systems* 13, no. 1 (2020): 914. https://doi.org/10.2991/ijcis.d.200615.002

[3] Das, Kajaree, and Rabi Narayan Behera. "A survey on machine learning: concept, algorithms and applications." *International Journal of Innovative Research in Computer and Communication Engineering* 5, no. 2 (2017): 1301-1309.

[4] Meerza, Syed Irfan Ali, Moinul Islam, and Md Mohiuddin Uzzal. "Q-learning based particle swarm optimization algorithm for optimal path planning of swarm of mobile robots." In *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, pp. 1-5. IEEE, 2019. https://doi.org/10.1109/ICASERT.2019.8934450

[5] Luo, Xiulian, Youbing Gao, Shao Huang, Yaodong Zhao, and Shengmiao Zhang. "Modification of Q-learning to adapt to the randomness of environment." In *2019 International Conference on Control, Automation and Information Sciences (ICCAIS)*, pp. 1-4. IEEE, 2019. https://doi.org/10.1109/ICCAIS46528.2019.9074718

[6] Arvind, C. S., and J. Senthilnath. "Autonomous RL: Autonomous vehicle obstacle avoidance in a dynamic environment using MLP-SARSA reinforcement learning." In *2019 IEEE 5th International Conference on Mechatronics System and Robots (ICMSR)*, pp. 120-124. IEEE, 2019. https://doi.org/10.1109/ICMSR.2019.8835462

[7] Somasundaram, Thamarai Selvi, Karthikeyan Panneerselvam, Tarun Bhuthapuri, Harini Mahadevan, and Ashik Jose. "Double Q-learning Agent for Othello Board Game." In *2018 Tenth International Conference on Advanced Computing (ICoAC)*, pp. 216-223. IEEE, 2018. https://doi.org/10.1109/ICoAC44903.2018.8939117

[8] Wang, Yan, Jie Tong, Tian-Yu Song, and Zhan-Hong Wan. "Unmanned surface vehicle course tracking control based on neural network and deep deterministic policy gradient algorithm." In *2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO)*, pp. 1-5. IEEE, 2018. https://doi.org/10.1109/OCEANSKOBE.2018.8559329

[9] Bakar, Mohamad Hafiz Abu, Abu Ubaidah Shamsudin, and Ruzairi Abdul Rahim. "Simulation of drone controller using reinforcement learning AI with hyperparameter optimization." In *2020 IEEE 10th International Conference on System Engineering and Technology (ICSET)*, pp. 167-172. IEEE, 2020.

[10] Myeong, W. C., Kwang Yik Jung, and Hyun Myung. "Development of FAROS (fire-proof drone) using an aramid fiber armor and air buffer layer." In *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pp. 204-207. IEEE, 2017. https://doi.org/10.1109/URAI.2017.7992713

[11] Karthik, P. B., Keshav Kumar, Vikrant Fernandes, and Kavi Arya. "Reinforcement learning for altitude hold and path planning in a quadcopter." In *2020 6th International Conference on Control, Automation and Robotics (ICCAR)*, pp. 463-467. IEEE, 2020. https://doi.org/10.1109/ICCAR49639.2020.9108104

[12] Cheng, Qiao, Xiangke Wang, Jian Yang, and Lincheng Shen. "Automated enemy avoidance of unmanned aerial vehicles based on reinforcement learning." *Applied Sciences* 9, no. 4 (2019): 669. https://doi.org/10.3390/app9040669

[13] Harwin, Laya, and P. Supriya. "Comparison of SARSA algorithm and temporal difference learning algorithm for robotic path planning for static obstacles." In *2019 Third International Conference on Inventive Systems and Control (ICISC)*, pp. 472-476. IEEE, 2019. https://doi.org/10.1109/ICISC44355.2019.9036354

[14] Sichkar, Valentyn N. "Reinforcement learning algorithms in global path planning for mobile robot." In *2019 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, pp. 1-5. IEEE, 2019. https://doi.org/10.1109/ICIEAM.2019.8742915

[15] Zamora, Iker, Nestor Gonzalez Lopez, Victor Mayoral Vilches, and Alejandro Hernandez Cordero. "Extending the openai gym for robotics: a toolkit for reinforcement learning using ros and gazebo." *arXiv preprint arXiv:1608.05742* (2016).

[16] Sutton, Richard S., and Andrew G. Barto. *Reinforcement learning: An introduction*. MIT Press, 2018.

[17] Pfiffner, Steffen. "Teach Ros Effectively." *The Construct*. January 24, 2023. https://www.theconstructsim.com/for-campus/.

[18] Zhao, Tianqu, and Hong Jiang. "Landing system for AR. Drone 2.0 using onboard camera and ROS." In *2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*, pp. 1098-1102. IEEE, 2016. https://doi.org/10.1109/CGNCC.2016.7828941