Modifiedment the Performance of Q-learning Algorithm Based on Parameters Setting for Optimal Path Planning

Noor H. Fallooh[1*], *Ahmed* T. Sadiq[2], *Eyad* I. Abbas[3] and *Ivan* A. hashim[4]

[1,3,4]Electrical Engineering Department, University of Technology-Iraq
[2]Computer Science Department, University of Technology-Iraq

Abstract. In engineering, the use of mobile robots to teach automatic control is becoming more common because of the interesting experiments that can be conducted with them. In this paper, a mobile robot that applies reinforcement learning in different scenarios is shown,to get rewards, the agent learns by acting in the environment. creating a balance between new information and our current understanding of the environment. In this way, the algorithm can be divided into two stages: the learning stage and the operational stage. In the first phase, the robot learns how to go from where it is to a known destination, it builds a learning matrix that is subsequently utilized during the operational stage using the rewards and environment data. In this paper, the algorithm was studied in terms of rapid learning for the mobile robot and reducing the process of repetition in learning by specifying the values of alpha (α) and gamma (γ) in a way that is appropriate for preserving the variance and differentiation between them. To evaluate the robot's adaptability to various dynamic situations, several simulated test scenarios were executed. In the testing situations, several target motion kinds and numbers of obstacles with various dynamicity patterns were used. The test scenarios illustrated the robot's adaptability to various settings.

1 Introduction

Mobile robots have a difficult difficulty when it comes to path planning in dynamic obstacle settings. Dynamic environments are ill-suited for traditional path planning algorithms like A* and RRT because they are unable to adjust to environmental changes. Since it enables the robot to learn to avoid obstacles in real time, reinforcement learning (RL) is a potential method for path planning in dynamic situations. For path planning in dynamic obstacle situations, a modified RL algorithm is proposed in this study. The Q-table is represented by the suggested algorithm, which also makes use of prioritized experience replay to increase learning efficiency. It was demonstrated that the suggested method could successfully plan pathways in dynamic obstacle settings when it was tested in a simulated setting [1]. All things considered, Q-learning is an effective and adaptable reinforcement learning algorithm that can be used to a number of different issues [2]. It is an excellent option for many applications because it is reliable, efficient, and model-free [3]. Because robotics offers a wide range of interconnected aspects, including mechanics, electronics, control, programming, and so forth, it is an ideal platform for teaching the principles of control engineering [1]. Robots, being self-governing machines, exemplify a closed control loop comprising sensors, actuators, and a control unit. For this reason, a lot of colleges use robots to teach fundamental control concepts [4]. Position control is one of the most fundamental tasks that mobile robots may carry out. Point stabilization or position control of a nonholonomic mobile robot is the name given to this experiment [5]. It addresses the challenge of moving a robot from its present location to a predetermined destination [6]. A robot that is mobile is one that has the ability to move around its surroundings [7]. It is managed by software that recognizes its environment and moves in a safe and regulated way using sensors and other technologies [8]. Applications for mobile robots are growing in popularity, ranging from personal to medical to industrial automation [9]. Numerous energy sources, such as batteries, fuel cells, and solar panels, are used to power mobile robots [10]. They usually have a variety of sensors to aid in their surroundings navigation, including radar, lasers, and cameras. To enable them to move, they also have an assortment of actuators, including wheels, tracks, and legs [4]. Mobile robots are going to becoming ever more powerful and adaptable as technology advances. There are numerous path planning algorithms, and each has advantages and disadvantages of its own. The path planning algorithms typical is:

* Corresponding author: Eee.21.14@grad.uotechnology.edu.iq

The well-known and effective path planning algorithm search: A* will always identify the shortest route from the starting point to the desired state. A* can, however, take a while to navigate through expansive or complicated landscapes.

Random trees that explore quickly (RRT): RRT is a probabilistic path planning system that finds a path quickly in complicated or large-scale environments. RRT does not, however, promise to locate the shortest way. Path planning algorithm that is easy to use and effective for settings that can be represented as a grid is called grid-based path planning [4].

Robot path planning performance can be impacted by a variety of additional variables in addition to path planning algorithms, including the complexity of the environment, the speed of the robot, and the quality of the map [11]. For many robotic applications, robot path planning is an essential component [12].The contribution of our work is to enhance the performance of Q-learning algorithm based on parameters setting, these parameters are (α) and (γ). The rest of paper including related works, robot path planning based on Q-Learning, enhancement the performance of Q-learning algorithm, experimental results and conclusion.

2 Related Works

Ee Soong Low, Pauline Ong, Kah Chun Cheah [13] in 2019, The results of the experimental evaluation of the proposed improved Q-learning in a tough environment with a distinct arrangement of obstacles indicate that, when Q-values are initialized suitably using the FPA, Q-learning can converge faster. Furthermore, the efficacy of the suggested algorithm is confirmed in an actual experiment utilizing a mobile robot with three wheels.

Hongqiang Sang, Yusong You at el [14] in 2021: It's becoming more common to use a formation fleet of unmanned surface vehicles (USVs). In order to plan the best routes and offer useful collision avoidance techniques for the efficient navigation of the USV fleet, formation path planning algorithms are crucial. The multiple sub-target artificial potential field (MTAPF), a unique deterministic algorithm based on an enhanced APF, is presented in this study in order to guarantee the optimality, rationality, and path continuity of the formation trajectories. An enhanced heuristic A* algorithm produces the global optimal path, which is referred to as the MTAPF in the local path planning algorithm.

Tian Ma, Jiahao Lyu at el [15] in 2022: In the area of mobile robot path planning, the (QL) algorithm has seen a rise in usage lately. The majority of the time, its selection policy is blind during the early stages of the search, which slows down the convergence of ideal solutions, particularly in complicated environments. In order to address these issues and guarantee the caliber of the intended path, a continuous local search Q-Learning (CLSQL) approach was suggested. First, separate local ecosystems are progressively created within the global environment. Next, each local environment's intermediate spots are searched using previous knowledge. The next step is to really search between each intermediate place and the destination point.

Talal Bonny, Mariam Kashkash [16] in 2022: suggested a brand-new, cutting-edge method for determining a mobile robot's best course in a two-dimensional environment. The Bees Algorithm (BA) in conjunction with the Q-Learning Algorithm is used to determine the best path. It is suggested to use a novel approach for creating the starting population, which may be used independent of the quantity and placement of environmental impediments. The BA's local search feature is implemented as Q-Learning. Finding the best course with fewer BA iterations is the goal of the hybridization of the Q-Learning with the BA. This approach uses Q-Learning's sterilization to discover the shortest path and the BA to solve the problem without any restrictions.

Qi Jiang1 [17] in 2021: In order to solve the path planning problem for mobile robots, researchers studied and created a Q-learning algorithm-based path planning method. This method relies only on the interaction between the robot's current state and surroundings to determine rewards and punishments for actions, which in turn determines the robot's next course of action. The original Q-learning algorithm had low efficiency and sluggish convergence. To address these issues, the system was updated so that the robot could finish planning rapidly and find the shortest and best path. The environment running the software to visualize the convergence process and collect data was established using the grid method.

Hyeok Soo Lee and Jong pil Jeong [18] in 2021: the application of reinforcement learning technology to automated logistics in a warehouse setting to optimize mobile robot routes. In order to determine the foundations needed for arranging a mobile robot's path and applying reinforcement learning techniques for path optimization, the outcomes of trials employing two fundamental algorithms are compared. The methods were put to the test in the identical experimental conditions and surroundings using a mobile robot's path optimization simulation. the improvement from the earlier experiment and carried out more tests to validate the improvement. The experimental findings contributed to our understanding of the traits and variations of the reinforcement learning algorithm.

3 Materials and Methods

3.1 Robot Path Planning Based on Q-Learning

A model-free reinforcement learning algorithm is called Q-learning. This implies that for the agent to learn, no prior knowledge of the environment is necessary. All the agent does is take lessons from its past experiences [19]. Maintaining a database of values, or Q-table, that associates states with actions is how Q-learning operates. The expected reward for

performing a certain action in a specific condition is represented by each entry in the Q-table [5]. Q-learning is a model-free reinforcement learning algorithm that determines an agent's best course of action in a Markov decision process (MDP). A transition function, reward function, discount factor, collection of states, and set of actions make up an MDP, which is a mathematical description of an environment [20].

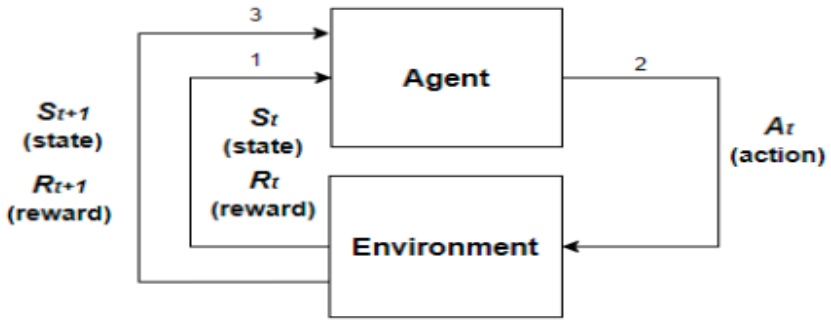


Fig. 1. The essential elements and interactions of Reinforcement Learning

The compensation the agent gets for changing from one state to another is decided by the reward function. The agent's value judgment for future benefits is based on the discount factor. A collection of state-action values known as a Q-table is updated iteratively as part of the Q-learning process. The anticipated value of performing an action in a specific condition is kept in the Q-table. The agent initially gathers information about the state-action values by investigating the surroundings. When the agent has gathered sufficient information, it starts to use the Q-table to determine what steps to take [21].

Q-learning is an efficient algorithm in terms of efficiency. It has the ability to quickly determine the best course of action for each given scenario. The some of the challenges of using Q-learning [22]:

- Q-learning needs to strike a balance between exploitation (doing the best it can) and exploration (doing new things). In complicated situations, this might be a difficult challenge to solve.
- Convergence: Q-learning might not always lead to the best course of action. This is because the technique is iterative and depends on random sampling.
- Requirements for data: Q-learning needs a large amount of data in order to determine the best course of action. In certain situations, such as when the environment is very big or the agent is unable to engage with it frequently, this can be difficult [23].

Q-learning is a potent reinforcement learning algorithm that can be used to a wide range of tasks, despite these difficulties. For scientists and engineers working on the creation of intelligent agents, it is an invaluable resource [4]. Q-learning is a reinforcement learning algorithm that learns to map from states and actions to expected rewards. The Q-table is a table that stores the expected rewards for each state and action. The Q-table is updated after each interaction with the environment, based on the reward received and the next state [24].

The following is the equation for updating the Q-table in Q-learning:

$$Q'(s,a) = Q(s,a) + \alpha(r + \gamma\,(max_{a'}\,Q(s',a'))) \qquad (1)$$

where: Q(s, a) :is the expected reward for taking action a in state s

r: is the reward received after taking action a in state s

s' :is the next state after taking action a in state s, alpha (α): is the learning rate gamma (γ): is the discount factor, $max_{a'}\,Q(s',a')$ is the maximum expected reward for taking any action a' in state s' [4]. The learning rate determines how often the Q-table is updated following each interaction with the environment [25]. A higher learning rate can lead to a faster update rate, but it may also result in overfitting the data; a lower learning rate can cause a slower update rate, but it may also cause the Q-table to converge more slowly [26]. The discount factor determines how much weight is assigned to future rewards; a higher discount factor can lead to a slower update rate, but it can also lead to a more heavily weighted consideration of future rewards. The maximum expected reward for taking any action a' in state s' is used to estimate the value of taking the best possible action in the next state [27]. This helps the Q-table to learn to take actions that lead to the most rewards in the long run [28]. Every time the agent interacts with the environment, the Q-table is updated until the agent finds a policy that maximizes the predicted reward. The mapping of states to actions that the agent will do to optimize the expected reward is known as the policy [29].The core idea of the QL algorithm is that when updating the Q value of a state-action pair, it uses

the Q value of the future state-action pair generated by the policy that will be evaluated instead of the Q value of the later state-action pair that follows the current policy. Specifically, the mobile robot randomly samples the environment and creates paths through multiple sampling in the path planning problem; during this process, the behavior policy and target policy iterate until the optimal course is found. The learning phase of the QL algorithm is described [4].in Algorithm 1

Algorithm 1:

The Q-learning Algorithm

1 Initialize $Q\ n \times m\ (s, a) = 0$ (n states and m actions)

2 Repeat

3 Using ε-greedy to select *a* from present state *s* ;

4 Take action *a* , *get r, s'* ;

5 Update *Q(s, a)* by (1)

6 $s \leftarrow s'$

7 Until *s* is destination

The idea of a value function, a function of a state that determines whether the environment an agent is placed in is good or bad, is incorporated into the reinforcement learning algorithm. Additionally, the policy behavioral method is used to define the value function. The probability of selecting potential courses of action in a state is its policy [4]. The state value function also took policy into account if the value function took into account both the cumulative reward and the state change simultaneously [20]. The predicted value of the return if policy π is followed from state s forward is the value function.

$$Q(s,a) = r(s,a) + max_a(Q(s',a)) \quad (2)$$

Equation (2) is generally converted into update Equation (3) by considering a factor of time

$$Q(s,a) \leftarrow Q(s,a) + \propto (r + \gamma max_{a'} Q(s',a') - Q(s,a)) \quad (3)$$

Q-Learning is improved by combining replanning algorithms with Q-Learning techniques. While the replanning algorithm randomly selects samples only from previously experienced information, the Q-Learning algorithm modifies a policy based on a real experience. The improvement of the Q-learning model obtains information for simulated experience through search control in the same way that it obtains information through modeling learning based on a real experience [30]. As in Q-Learning (3) [25]

3.2. Enhancement of Q-learning by controlling the parameters of algorithms

This study reviews the application of reinforcement learning as a path optimization strategy in a maze setting, and it provides experimental confirmation of how to reduce inactive exploration with a straightforward technique to maintain path search time and enhance path search accuracy. In order to evaluate path navigation in a complex maze setting and the advancement of Q-learning algorithms for reinforcement learning, we constructed a simulation environment. Next, we made a comparison between the experimental results regarding the final path and the path search periods. We used the Q-Learning method in a number of trials to try to improve the results, but we discovered that it is not very effective at discovering search paths in a complicated environment with plenty of barriers. The ability to randomly find a path is a feature of the Q-Leaning algorithm. To minimize this random path finding trial, the target position was chosen for movement, and the set reward value was high around the target position,' therefore the performance and efficiency of the agent improved with 'dynamic reward of improvement-Q-learning on action'. Agents generally investigate areas with high payoff values. The improvement Q-learning update equation is as (4).

$$Q(s,a) \leftarrow Q(s,a) + \propto '(r + (\gamma')maxQ(s',a') - Q(s,a)) \quad (4)$$

The improvement of Q- learning algorithm store past experiences in memory to generate the next state and a more accurate reward value.

4 Experiments and Results:

4.1 Maze Simulation Environment:

Referring to the research [26], we constructed a virtual experiment environment based on the previously stated maze environment in order to build a simulation environment. Obstacles in a maze setting include okay, start, and goal states. A 16 × 16 grid plan had several obstacles in it as shown in figure (3). A mobile robot was simulated moving from the starting location to the destination. In order to avoid obstacles at the starting point where a mobile robot was defined, the experiment's goal was to determine the best route to the target place. never changing course and never attempting to find a way from one place to another to reach a goal. Depending on the action value, move to the right, left, up, or down for each terminal condition as you approach obstacles. Extensive simulation experiments were conducted in order to train the robot and demonstrate the efficacy of the novel approach. Software from MATLAB was used to implement this simulation. Various settings were put into place for various circumstances, and the outcomes of these various states were utilized to evaluate how well the suggested Q-learning method performed.

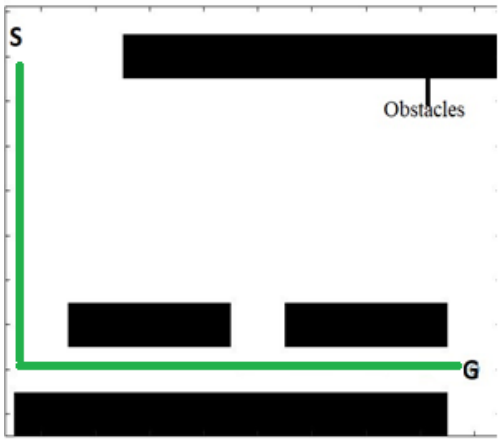
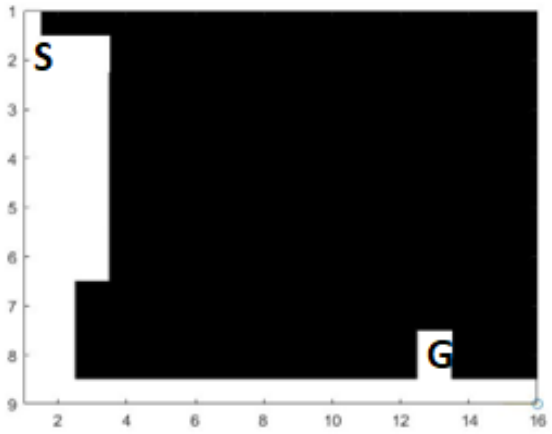
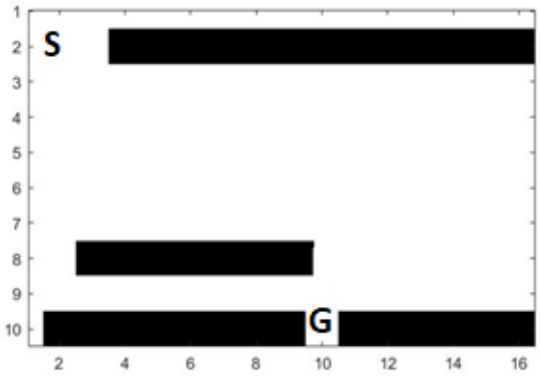


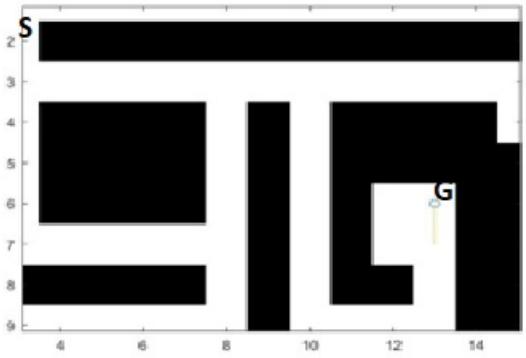
Fig. 2. The simple maze environment with start and goal position



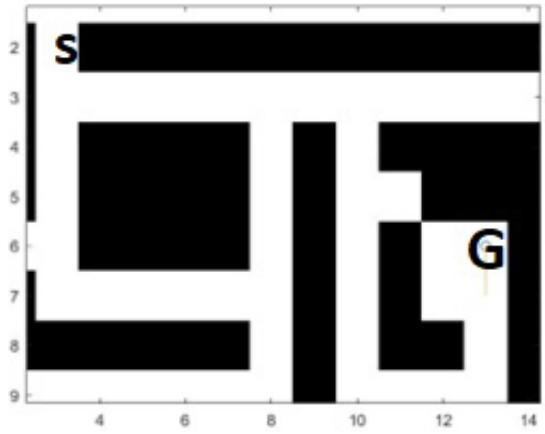
(a)



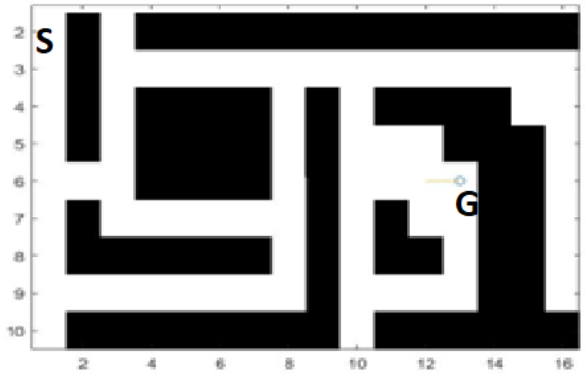
(b)



(c)



(d)



(e)

Fig. 3. shows the target position for the different environment a) close b) open c) randomly d) zigzag close e) zigzag open environment.

The two stages of the simulation are the training phase and the testing phase. During the training phase, different target and obstacle behaviors were applied in different scenarios for different conditions. As the robot navigates through each scenario's many states, it is learning. The previously discussed Q-table serves as a clearinghouse for the knowledge gathered from the various scenarios. After the learning phase, the robot is exposed to a more difficult environment before the testing phase commences.

4.2. Experimental Results

There are more hurdles in the testing situations some of them are dynamic, others of them static. The method was examined through a few testing scenarios for special, everyday situations such as the local minima problem. Random scenarios were used for some more testing. The test parameter value that used in different state simulation environment as shown in figure (4) of Q-learning and improvement Q-learning. The Learning Rate (α) and Discount Rate (γ) with improvement discount rate that make depending on the learning rate and obtained the result as shown in table (1)

Table 1. The result with Learning Rate (α) and Discount Rate (γ) with the number of obstacles in open environment is 33.

Discount Rate (γ)	**Time in sec. with 0.1**	**Path of mobile robot**	**Time in sec. with 0.2**	**Path of mobile robot**	**Time in sec. with 0.3**	**Path of mobile robot**
1	1.691	Without smoothing	1.566	Without smoothing	1.569	winding path
0.95	1.689	Without smoothing	1.577	Without smoothing	1.587	winding path
0.9	1.775	smoothing	1.524	Without smoothing	1.470	Without soothing
0.8	1.756	smoothing	1.5387	smoothing	1.466	Without soothing
0.7	1.882	Less smoothing	1.551	Less smoothing	1.505	smoothing
0.6	1.999	winding path	1.731	winding path	1.531	Without soothing
0.5	2.017	winding path	1.637	winding path	1.564	winding path
0	long time	Critical smoothing	Long time	Critical smoothing	Long time	winding path

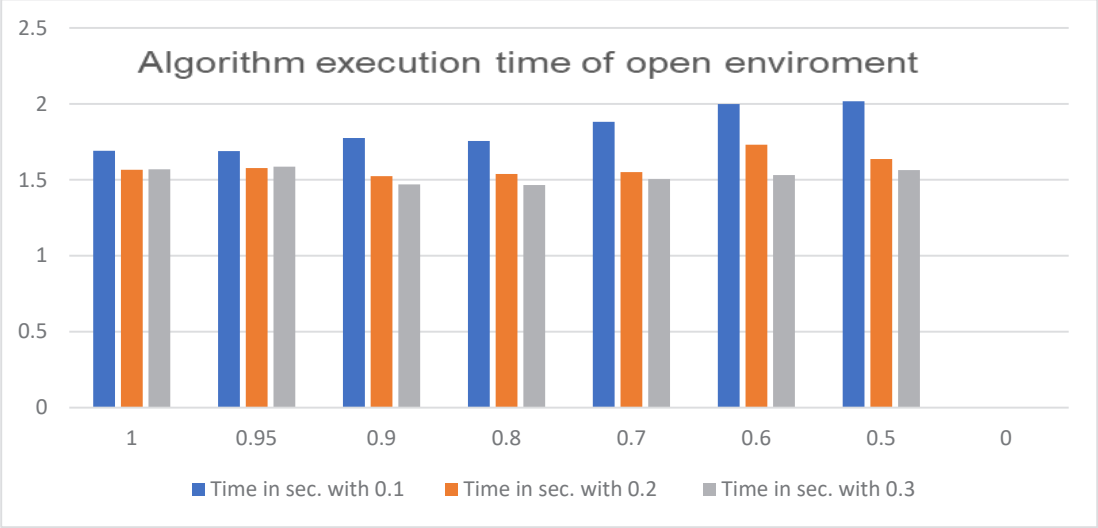
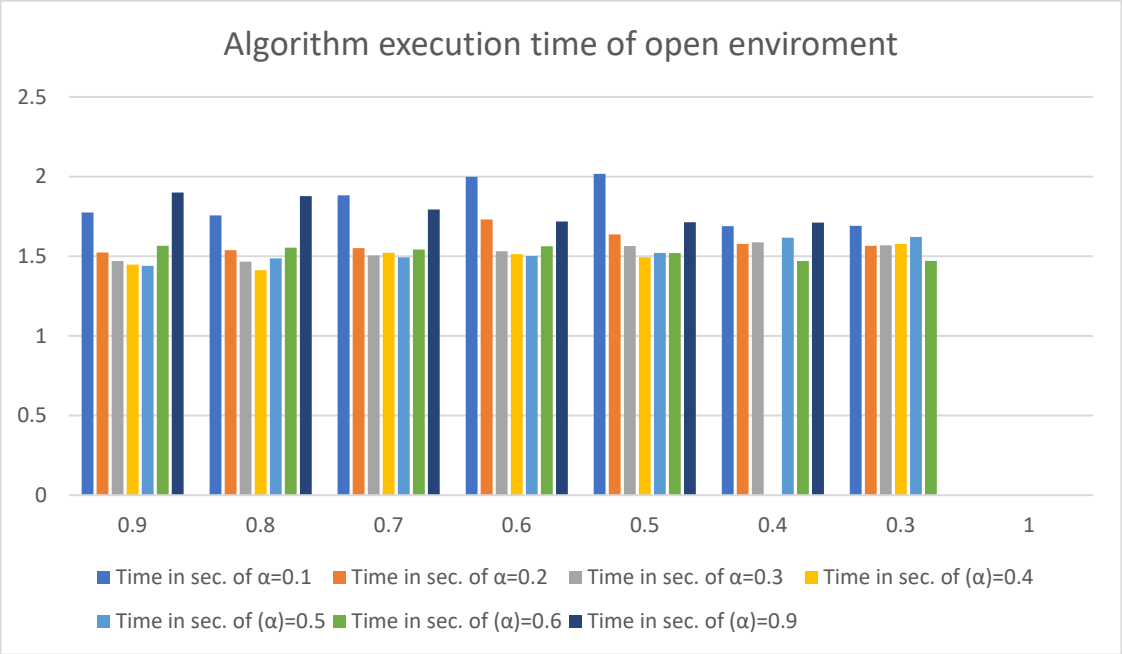


Table 2. take different Learning Rate (α) and Discount Rate (γ) with the number of obstacles in open environment is 33

Discount Rate (γ)	**Time in sec. of α=0.1**	**Time in sec. of α=0.2**	**Time in sec. of α=0.3**	**Time in sec. of (α)=0.4**	**Time in sec. of (α)=0.5**	**Time in sec. of (α)=0.6**	**Time in sec. of (α)=0.9**	**Time in sec. of (α)=1**
0.9	1.775	1.524	1.470	1.474	1.540	1.566	1.9	5 h.
0.8	1.756	1.5387	1.466	1.512	1.686	1.754	1.877	6 h.
0.7	1.882	1.551	1.505	1.522	1.593	1.642	1.793	5.30 h.
0.6	1.999	1.731	1.531	1.533	1.581	1.663	1.718	12 h.
0.5	2.017	1.637	1.564	1.593	1.521	1.520	1.713	5.35 h.
0.4	1.689	1.577	1.587	1.647	1.616	1.470	1.711	10 h.
0.3	1.691	1.566	1.569	1.577	1.621	1.471	1.612	12.48 h.
1	5.36 h.	5.21 h.	6 h.	6 h.	7 h.	7 h.	7 h.	15 h.

The results in the table (2) we can notice that the smoothing of the path changes according to the difference between α and γ, when α = 0.4 and γ = 0.8 and 0.9 obtaining results is winding path between the start position and target position leads to consumption of the mobile robot's battery but the time of execution the algorithm smallest equal to 1.474 sec. at α=0.4 and time =1.512 at α= 0.5 this good to complete the searching of best path at good time . when α = 0.6 and γ = 0.1 to 0.9 obtaining results without smoothing path between the start position and target position leads to more consumption of the mobile robot's battery. The best result of smoothing path is obtained at α= 0.1 and γ= 0.9, α= 0.2 and γ= 0.8 ,α= 0.4 and γ= 0.4, α= 0.5 and γ= 0.5, with the time of consumption the algorithm as shown in table (2) .



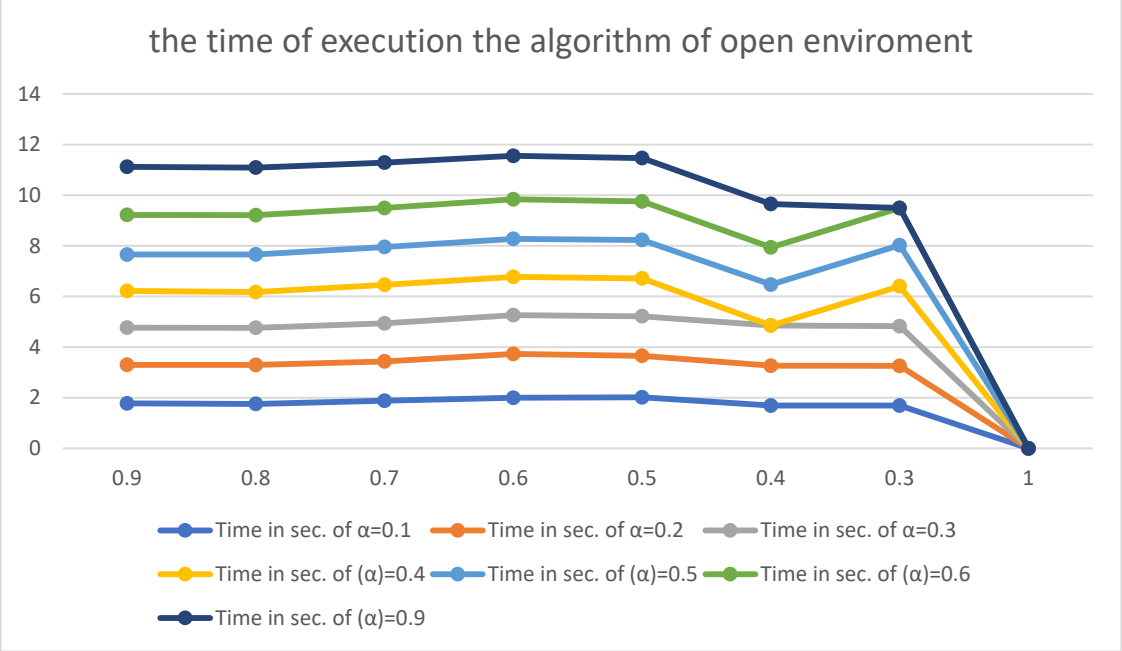


Table 3. take different Learning Rate (α) and Discount Rate (γ) with the number of obstacles in close environment is 102.

Discount Rate (γ)	**Time in sec. of (α)=0.2**	**Time in sec. of (α)=0.4**	**Time in sec. of (α)=0.5**	**Time in sec. of (α)=0.6**	**Time in sec. of (α)=0.9**	**Time in sec. of (α)=1**
0.95	2.280	2.325	2.346	2.237	2.232	Long time

0.9	2.252	2.328	2.348	2.186	2.236	Long time
0.8	2.314	2.398	2.471	2.172	2.396	Long time
0.7	2.257	2.461	2.496	2.218	2.367	Long time
0.6	2.358	2.463	2.583	2.209	2.356	Long time
0.5	2.369	2.455	2.563	2.299	2.366	Long time
0.4	2.401	2.462	2.568	2.298	2.419	Long time
0.3	2.950	2.477	2.684	2.272	2.402	Long time
0.1		2.523	2.672	2.311	2.410	Long time

The results in the table (3) we can notice that the smoothing of the path changes according to the difference between α and γ, that is give the winding path at α= 0.2, 0.4 and 0.6 at γ= 0.8, 0.9 and 0.95. when (α= 0.4,0.5,0.6 and γ=0.3,0.1) and (α=0.5 and γ=0.8,0.9) and (α=0.9, γ=0.5,0.4,0.3,0.1) is obtained the path without smoothing. Is giving the smoothing path at (α=0.2, γ=0.8), (α=0.4, γ=0.6, (α=0.5, γ=0.5), (α=0.6, γ=0.4).

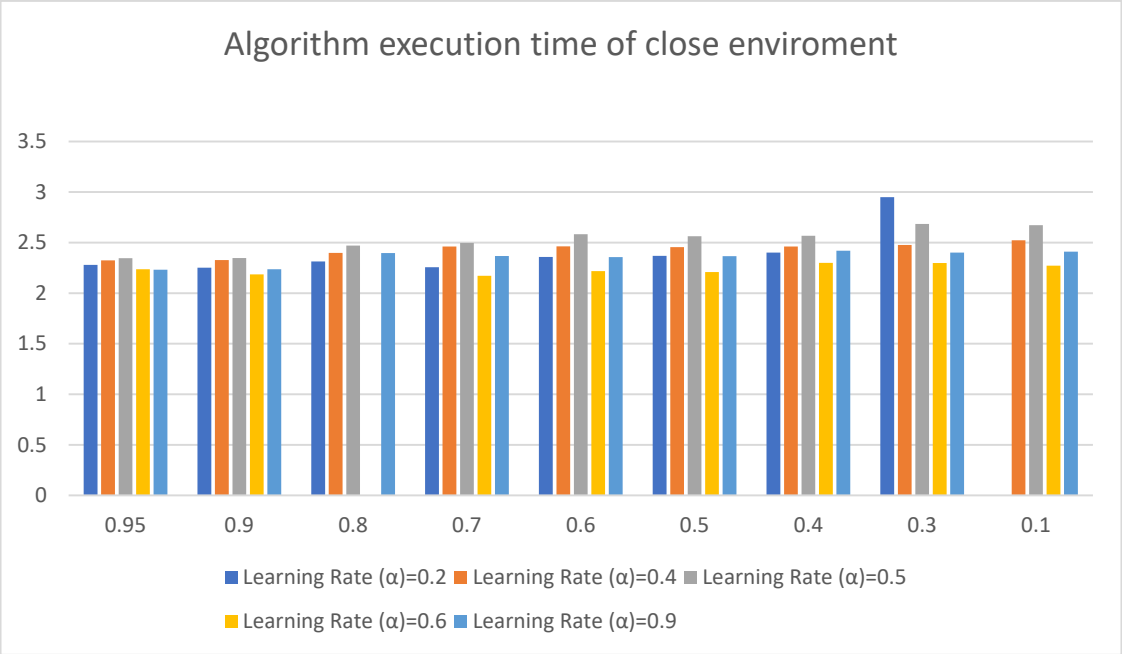
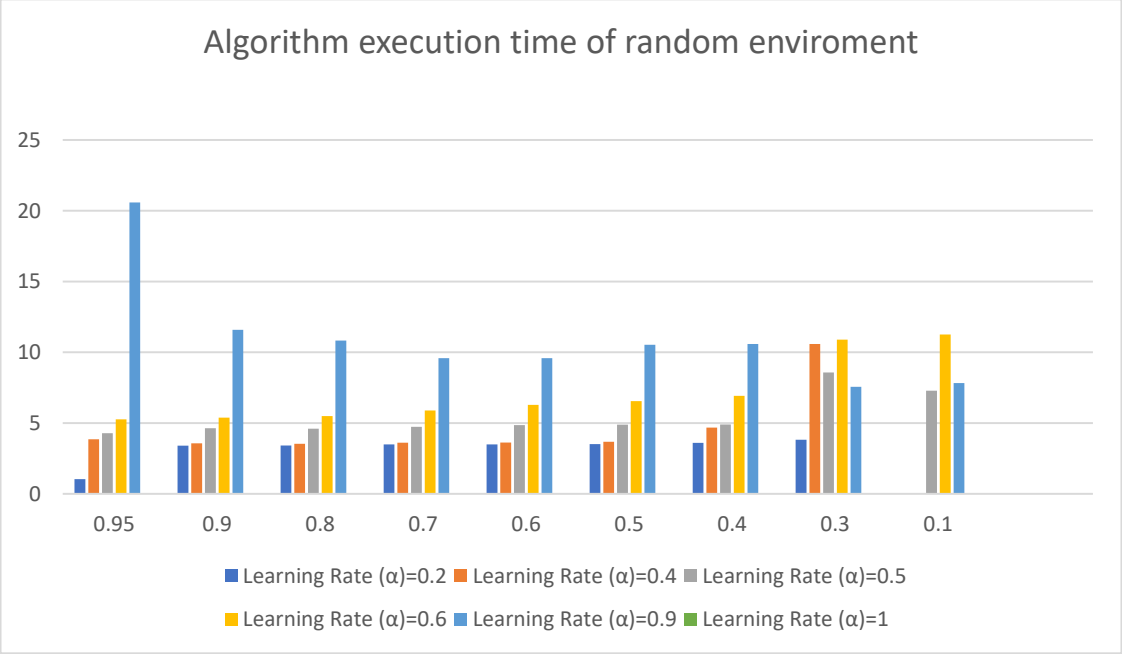


Table 4. take different Learning Rate (α) and Discount Rate (γ) with the number of obstacles in randomly environment is 77.

Discount Rate (γ)	**Time in sec. of (α)=0.2**	**Time in sec. of (α)=0.4**	**Time in sec. of (α)=0.5**	**Time in sec. of (α)=0.6**	**Time in sec. of (α)=0.9**	**Time in sec. of (α)=1**
0.95	3.401	3.856	4.281	5.261	20.589	

0.9	3.412	3.568	4.639	5.386	11.583	Long time
0.8	3.490	3.537	4.607	5.493	10.833	Long time
0.7	3.495	3.618	4.729	5.894	9.581	Long time
0.6	3.510	3.628	4.852	6.283	9.586	Long time
0.5	3.603	3.682	4.892	6.548	10.527	Long time
0.4	3.827	4.682	4.894	6.926	10.589	Long time
0.3	Long time	10.586	8.573	10.893	7.562	Long time
0.1	Long time	Long time	7.286	11.253	7.824	Long time

The results in the table (4) we can notice that the smoothing of the path changes according to the difference between α and γ, that is give the winding path at α= (0.2, 0.5 and γ=0.9 and 0.95). when (α= 0.2and γ=0.6,0.5 and 0.4) is obtained the path without smoothing. Is giving the smoothing path at (α=0.2, γ=0.8), (α=0.4, γ=0.6, (α=0.5, γ=0.5), (α=0.6, γ=0.4).



The path search time of the Q algorithm was times faster than that of the improvement Q-learning algorithm, and the improvement-Q-learning algorithm's final path was more smoothed and shorter than that of the Q-Learning algorithm when compared to the average of the results of the several experiments. Consequently, the path search time and the final path length produced by the Q-Learning and improvement-Q-learning algorithms' experiments yielded disappointing results.

4 Conclusions

Further testing revealed that the final path length was improved to the same degree as the improvement Q-learning method and that the path search time was marginally faster than the Q-Learning technique. Achieving both high path search accuracy and minimal path search time is a challenging objective. The learning process may take too long if the learning speed is set too late, or it may not converge to the optimal value function if it is set too quickly. This fundamental study is an illustration of how to improve the surroundings or the movement of the target or obstacles. A new concept of the state space was developed in order to enable the application of this method in a dynamic context. The goal of this revised definition is to approximate how humans might think in situations like these. The quicker learning to acquire the high reward and disregard the low reward values to avoid bumping into obstacles occurs when the gamma values are closer to one. Furthermore, the learning rate will increase with alpha values that are closer to zero. The path planning must be smoothing to keep the batterie's of mobile robot to more time and the consumption time of algorithms is less than the path of mobile robot without smoothing or winding path. We have to keep the alpha (α) and gamma (γ) differences intact in order to prevent leaving the shortest path in the path plan for various situations.

References

1. P. Wang, Ch. Chan and A. de-L. Fortelle, A Reinforcement Learning Based Approach for Automated Lane Change Maneuvers, 2018 IEEE Intelligent Vehicles Symposium (IV), China, 2018.
2. M. Naeem, S.T.H. Rizvi, and A. Coronato, A Gentle Introduction to Reinforcement Learning and Its Application in Different Fields", IEEE ACCess 8, 2020.
3. G.U.O. Tong, N. Jiang, L.I. Biyue, Z.H.U. Xi , Y. Wang , UAV navigation in high dynamic environments: A deep reinforcement learning approach, Chinese Journal of Aeronautics, Production and hosting by Elsevier 2020 .
4. Parag Kulkarni, "Reinforcement and Systemic Machine Learning for Decision Making", Published by John Wiley & Sons, Inc., Hoboken, Published simultaneously in Canada (IEEE series on systems science and engineering; ISBN 978-0-470-91999-6.
5. Marco Wiering and Martijn van Otterlo (Eds.),' Reinforcement Learning State-of-the-Art', ISSN 1867-4534, ISBN 978-3-642-27644-6, DOI 10.1007/978-3-642-27645-3, Springer
6. C. Wang, J. Wang, Y. Shen, Autonomous navigation of UAVs in large-scale complex environments: A deep reinforcement learning approach. IEEE Trans Veh Technol 68,3, (2019) 2124–36.
7. L. Chang, L. Shan, Ch. Jiang, and Y. Dai, Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment, Autonomous Robots, Springer Science+Business Media, LLC, part of Springer Nature 2020.
8. J. Kim, M. Hong, K. Lee, D.W. Kim, Y.-L. Park, and S. Oh, Learning to Walk a Tripod Mobile Robot Using Nonlinear Soft Vibration Actuators with Entropy Adaptive Reinforcement Learning,IEEE Robotics And Automation Letters 2020.
9. S.S. Mousavi, M. Schukat, and E. Howley, Traffic light control using deep policy gradient and value-function-based reinforcement learning, IET The Institution of Engineering and Technology 2017.
10. F.A. Raheem , A.T. Sadiq, N. A. F. Abbas, Robot Arm Free Cartesian Space Analysis for Heuristic Path Planning Enhancement", International Journal of Mechanical & Mechatronics Engineering IJMME-IJENS 19(01) 2019.
11. A.T. Sadiq, and A. H. Hasan, Robot Path Planning Based on PSO and D* Algorithms in Dynamic Environment, International Conference on Current Research in Computer Science and Information Technology (ICCIT), Slemani – Iraq 2017.
12. Yun-HuaWua,1, Zhi-ChengYua, Chao-YongLib, Meng-JieHea, BingHuaa, Zhi-MingChena, 'Reinforcement learning in dual-arm trajectory planning for afree-floating space robot', Aerospace Science and Technology www.elsevier.com/locate/aescte. 2021
13. Ee Soong Low, Pauline Ong , Kah Chun Cheah, Solving the optimal path planning of a mobile robot using improved Q-learning', Robotics and utonomous Systems journal homepage: www.elsevier.com/locate/robot , in 2019.
14. H. Sang, Y. You, X. Sun, Y. Zhou, and F. Liu, the hybrid path planning algorithm based on improved A* and artificial potential field for unmanned surface vehicle formations, Ocean Engineering journal 2021.

15. T. Ma, and J. Lyu, J. Yang, R. Xi, Y. Li, J. An, and Ch. Li, CLSQL: Improved Q-Learning Algorithm Based on Continuous Local Search Policy for Mobile Robot Path Planning 22 (2022)5910.
16. Talal Bonny and Mariam Kashkash, 'Highly optimized Q-learning-based bees' approach for mobile robot path planning in static and dynamic environments', J Field Robotics. 2022; 39:317–334.4, wileyonlinelibrary.com/journal/rob, 2022.
17. J. Qin, X. Han, and G. Liu et al," Path Planning Method of Mobile Robot Based on Q-learning, Journal of Physics: Conference Series,2022.
18. H.S. Lee, and J. Jeong, Mobile Robot Path Optimization Technique Based on Reinforcement Learning Algorithm in Warehouse Environment, Appl. Sci.11(2021)1209. https://doi.org/10.3390/ app11031209
19. M.E. Aloud, and N. Alkhamees, Intelligent Algorithmic Trading Strategy Using Reinforcement Learning and Directional Change", date of current version August 24, 2021. http://doi.org/ 10.1109/ACCESS.2021.3105259.
20. X.B. Peng,M. Andrychowicz, W. Zaremba and P. Abbeel, Sim-to-real transfer of robotic control with dynamics randomization. In: International Conference on Robotics and Automation. IEEE 2018.
21. A. Nair, B. McGrew, M. Andrychowicz, W.Zaremba and P. Abbeel, Overcoming exploration in reinforcement learning with demonstrations". In: International Conference on Robotics and Automation. IEEE (2018) 6292–6299.
22. T. Johannink , S. Bahl , A. Nair et al., Residual reinforcement learning for robot control. In: International Conference on Robotics and Automation. IEEE 2019.
23. R.S. Sutton, and A.G. Barto, Introduction to Reinforcement Learning, 2nd ed.; MIT Press: London, UK (2018) 1–528
24. B. Jang,M. Kim, G. Harerimana, J.W. Kim, Q-Learning Algorithms: A Comprehensive Classification and Applications. IEEE 7 (2019) 133653–133667
25. W.-G. Han, S.M. Baek, and T.Y. Kuc, Genetic algorithm-based path planning and dynamic obstacle avoidance of mobile robots. In Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Orlando, FL, USA 1997.
26. N. Kohl, and P. Stone, Policy gradient reinforcement learning for fast qudrupedal locomotion. In: International Conference on Robotics and Automation. IEEE 2004.
27. M. Kirtas, K. Tsampazis, N. Passalis, A. Deepbots, A Webots-Based Deep Reinforcement Learning Framework for Robotics. In Proceedings of the 16th IFIP WG 12.5 International Conference AIAI 2020, Marmaras, Greece, (2020) 64–75
28. V.N. Sichkar, Reinforcement Learning Algorithms in Global Path Planning for Mobile Robot. In Proceedings of the 2019 International Conference on Industrial Engineering Applications and Manufacturing, Sochi, Russia 2019.
29. F.A. Raheem, A.T. Sadiq, N. A. F. Abbas, Optimal Trajectory Planning of 2-DOF Robot Arm Using the Integration of PSO Based on D* Algorithm and Cubic Polynomial Equation", the first for Conference engineering researches 2017.
30. L. Jiang, R. Wei and D. Wang, Multi-UAV Roundup Inspired by Hierarchical Cognition Consistency Learning Based on an Interaction Mechanism, Drones 7(2023) 462.
31. M.A.K. Jaradat , M. Al-Rousan , L. Quadan, Reinforcement based mobile robot navigation in dynamic environment", Robotics and Computer-Integrated Manufacturing 27 (2011).