Q-Learning in a Dynamic Environment containing obstacles with random movement

Author Names Omitted for Anonymous Review

***Abstract*— Reinforcement learning is the science of decision making. It facilitates the agent to learn an optimal, or nearly-optimal, policy that maximizes the "reward function" or other user-provided reinforcement signal that accumulates from the immediate rewards. This work describes a unique way of using Q-Learning for dynamic environments where movement of obstacles is random and the robot has to find a way to reach the Goal. In this work we use combination of Q-Learning algorithm and Neighbour searching algorithm for obstacle avoidance and proper navigation towards the Goal. Evaluation for this work is done by comparing the output of this work with pre-exsiting algorithms in dynamic environments and probability density comparisons.**

I. INTRODUCTION

This paper will deal with Q-Learning implementation on a robot in a environment where the obstacles are placed randomly. The environment used for this project is a dynamic environment where the obstacles move and their movement is also randomized. Q learning is implemented, along with the use of Bellman Equation another approach is implemented along with this which governs the Q-Table used for learning by robot.

This approach checks if there is any obstacle in the vicinity of the robot and based on the sensing and input received it updates the Q-Table based on the input received either by assigning a positive reward or negative depending on the vicinity of the robot. The main problem considered here is to able to implement Q-Learning in a dynamic environment. These types of problems have been researched upon a large period of time and they are important in applications as the navigation or obstacle avoidance algorithms which are already developed work noteworthy but they were developed for static environments.

Q-Learning being also used mainly in static environments and years of research implementing this method on dynamic environment has huge real world applications as in real world the environments rather dynamic.The environment used in this work is a Grid-World whose grid is defined as a 10x10 grid. There are 20 obstacles present in the environment with robot at one corner of the grid and goal on the opposite diagonal edge. In this environment there are 20 obstacles present at random locations at the start and they move randomly when the robot takes a step in the environment. The desired output attainable from this project is for the robot to be able to navigate through the environment and reach goal maximum number of times in a given set of tests.

II. PRIOR WORK

Due to the complexity of interactive environments, dynamic obstacle avoidance path planning poses a significant challenge to agent mobility. Dynamic path planning is a complex multi-constraint combinatorial optimization problem. Some existing algorithms easily fall into local optimization when solving such problems, leading to defects in convergence speed and accuracy. Reinforcement learning has certain advantages in solving decision sequence problems in complex environments. (I was unable to add and cite the references to the prior work, due to which this section is mostly incomplete. For this work there is lot or relevant or analogic prior work and I will put those once I figure out the citation issue.)

III. PROPOSED METHOD

In this work the main proposed method is of combining Q-Learning algorithm for updating the Q-Table but instead of making this a static function we implement time as a function for the Q-Table and let the optimal solution be calculated from the average of all outputs obtained from training of this model. Furthermore, we also make Q-Table a function of the neighbouring cells such that if there is no obstacle in the neighbouring cells then it will maximize the reward for movement, thereby encouraging robot for exploration in order to navigate to goal. This is the Neighbour searching algorithm. In the end we have a Q-Table which is a function of 2 inputs and is trained upon, tested under random obstacles and, validated.

IV. VALIDATION

Validation will done by testing the trained model and recording the number of successful test completed. These results will be interlinked and discussed for distribution of the range between the Goal and Robot vs Successful Navigation Probability. Furthermore, the output of this model will be compared against other existing algorithms to obtain further insights on this model.

V. CONCLUSION

The future work will include implementing large neural networks on this model to further optimize and predict the path navigation through rigorous exploitation, resulting in further optimising this model and making it efficient and faster.