The Achievement of Dynamic Obstacle Avoidance Based on Improved Q-Learning Algorithm

Qixuan Hao

School of Electromechanical and Information Engineering, China University of Mining and Technology (Beijing), Beijing, China

2010490407@student.cumtb.edu.cn

Abstract. Dynamic obstacle avoidance is a classic problem in robot control, which involves the ability of a robot to avoid obstacles in the environment and reach its destination. Among various path planning algorithms, the dynamic obstacle avoidance issue may be resolved using the reinforcement learning algorithm Q-learning. This article provides a comprehensive review of the recent research progress and achievements in the field of dynamic obstacle avoidance, through the analysis and improvement of the Q-learning algorithm. The article begins by introducing the background and research status of dynamic obstacle avoidance, followed by a detailed exposition of the principles and implementation of the Q-learning algorithm. Subsequently, the shortcomings of the Q-learning algorithm are analyzed, and several improvement measures are proposed, such as combining deep learning with Q-learning, and using recombination Q-learning. Finally, the article summarizes the current application status of the Q-learning algorithm in dynamic obstacle avoidance and proposes future research directions.

Keywords: Q-learning algorithm, Dynamic obstacle avoidance, Improved Q-Learning Algorithm.

1. Introduction

The forward advancement of technology has ensued in an adding demand for automation and intelligence. Among the various fields in which artificial intelligence (AI) is making significant progress, the area of autonomous driving has been particularly noteworthy. By using AI, vehicles can be operated with reduced labor costs while improving efficiency and safety, which makes it a vital area of research. One of the key issues in this field is how to optimize routing and obstacle avoidance in complex and dynamic environments [1].

Path planning algorithms are a crucial component of autonomous driving systems. Common path planning algorithms include Dijkstra, Rapidly-exploring Random Trees (RRT), A*, and ant colony optimization algorithms [2, 3]. These algorithms can achieve path planning in continuous or discrete spaces, However they are inappropriate for practical applications because to their complex computational requirements and subpar real-time performance. Machine learning technology has become a viable answer to these problems in recent years.

Because to its broad use in a number of industries, including digital animation, gaming, customized recommendations, and autonomous driving, Watkins' Q-learning algorithm has drawn a lot of attention [4]. By learning from states and actions, this algorithm can quickly find the shortest path and possesses natural advantages compared to traditional path planning algorithms. Q-learning can better adapt to complex and uncertain environments.

In practical applications, Q-learning can be combined with traditional path planning algorithms to form a more comprehensive path planning method [5]. Researchers are exploring ways to improve the performance of the algorithm by integrating deep learning, reinforcement learning, and other technologies with path planning. These new technologies can more accurately simulate complex real-world environments and learn better path planning strategies, bringing greater hope for the application of autonomous driving in various fields.

Despite the promising developments in AI technology, the safety and privacy of autonomous driving remain a significant concern. There is a need for strengthening relevant laws and regulatory systems to address these issues. It is essential to ensure that autonomous driving technology is

developed and regulated in a manner that ensures the safety of all road users. The regulatory systems must be developed in such a way that they take into account the rapidly evolving nature of AI technology and its potential applications.

The rapid development of AI technology is leading to significant changes in the transportation industry. Autonomous vehicles have the potential to revolutionize how people move and interact with each other. The benefits of autonomous driving include increased safety, reduced traffic congestion, and reduced carbon emissions. However, the development and deployment of autonomous driving technology must be carried out in a responsible manner. The ethical implications of AI technology must be considered and addressed to ensure that the benefits are shared equitably and that the technology serves the greater good.

In conclusion, AI technology has enormous potential in the field of autonomous driving. The development and deployment of autonomous driving technology should be carried out in a manner that addresses the significant concerns around safety and privacy. Integration of AI technology with traditional path planning algorithms has already shown promising results, and further research in this area can lead to the development of more effective and efficient autonomous driving systems. The rapid evolution of AI technology is creating new opportunities and challenges, and it is important to work collaboratively to develop AI technology that serves the greater good.

2. Traditional Q-Learning Algorithm

A value-based supervised learning method called Q-learning seeks to learn the best course of action for a given environment so that the agent can maximize cumulative rewards. The agent uses the Q-learning process and keeps a Q-table, where Q(m,a) denotes the value of performing action an in state s, or the predicted return value for action an in the current state s. In order to get the best policy, the deputy continuously interacts with the surroundings to update the value function in the Q-table.

The algorithm's update rule is:

$$\mathrm{Q}(mi, At) \leftarrow Q(mi, At) + \beta[Rt+1+\xi \max_{\beta} Q(mi+1, \beta) - Q(mi, At)] \quad (1)$$

The renovate rule of the Q-learning algorithm is as follows: where mt symbolizes the current state, at represents the current action taken, Rt+1 represents the instant benefit obtained after taking action At, mt+1 represents the ensuing state entered, β stands for learning rate, another ξ is on behalf of present value. The expected long-term reward for every feasible actions an of the current state m t is calculated by taking the weighted average using the discounted amount. The term max a Q(mt+1, a) indicates the highest possible reward for all practical acts in the following state mt+1 [6].

To be able to choose actions, the Q-learning algorithm employs a greedy strategy, or selecting the behavior with the highest value as it stands at the moment. After the agent does an action, the utility function in the Q-table is modified based on the feedback provided by the environment. The agent constantly engages with the environment during training until the Q-values convergence or a predetermined attained the desired amount of iterations [7].

3. Improved Q-Learning Algorithm

3.1. Improvements for Basic Direction

To address the limitations of the Q-algorithm in dynamic obstacle avoidance, researchers have proposed several improvement methods as follows:

(1) Enhancing the state and behavior spaces' dimensionality. To more accurately describe the environment and the agent's state, state and behavior spaces are multi modal. can be increased. For example, in dynamic obstacle avoidance problems, factors such as the position and velocity of obstacles can be added as part of the state.

(2) Introducing a reward function. To better guide the agent's learning, a reward function can be introduced to help the agent understand the environment. For example, in dynamic obstacle avoidance problems, a reward function can be designed to encourage the agent to avoid obstacles as quickly as possible, while penalizing collisions or overly risky actions.

(3) Combining deep learning algorithms. In order to produce more precise Q-value functions, deep learning algorithms can automatically extract showcases from the state and behavior spaces. For instance, the Deep Q-Network (DQN) approach circumvents the issue of vast state and behavior spaces in the Q-learning process by estimating the Q-value function utilizing deep neural networks.

(4) Improving the balance between exploration and exploitation. To better balance exploration and exploitation, diverse strategies can be used. In some cases, exploration probability can be increased to allow the agent to try new actions, while in other cases, exploration probability can be decreased to improve decision-making accuracy.

(5) Combining reinforcement learning and planning algorithms. Reinforcement learning algorithms can learn the optimal policy, while planning algorithms can generate the optimal path through search algorithms. Combining the two algorithms can better solve the problem of dynamic obstacle avoidance.

Increasing the state and behavior spaces are multi modal can significantly improve the agent's decision-making accuracy. Introducing a reward function can help the agent better understand the environment and quickly avoid obstacles. Combining deep learning algorithms can improve the efficiency and accuracy of the agent's learning. The agent can better balance exploration and exploitation by adjusting the equilibrium between them. Algorithms for planning and reinforcement learning can be combined to more effectively address the issue of dynamic collision avoidance.

In addition, it is being noted that the use of these improvement methods must be carefully considered and implemented. Depending on the specific problem, different methods may be more or less effective. Therefore, further research and experimentation are necessary to determine the best approach for each problem. Ultimately, the goal is to create safe, efficient, and reliable autonomous driving systems that can improve transportation and benefit society.

3.2. Improvements for Dynamic Obstacle Avoidance

3.2.1 Sequence of State-Action-Reward-Next State-Next Action (SARSA) Algorithm

When updating the Q values, SARSA algorithm uses the behavior consider the situation as-is, rather than the maximum action in the next situation [8]. This makes SARSA algorithm more stable, but may lead to less accurate Q-value functions. In addition, SARSA algorithm has an important advantage, it can be applied to tasks that require continuous decision-making.

Deep reinforcement learning methodology is used to multi-UAV route planning and obstacle avoidance as a way to accomplish route planning and obstacle avoidance for UAVs in dynamic situations [9]. The algorithm takes the UAV's state, environmental perception data, and the UAV's actions as inputs, and learns and makes decisions through a neural network, ultimately outputting the UAV's actions to achieve desired functionality. The UAV adjusts its movement trajectory in a timely manner based on changes in the environment to avoid collision with obstacles or failure of action due to cumulative errors. This process involves the following steps:

(1) Environmental Perception: The UAV obtains real-time information about the surrounding environment through sensors and other devices. The perceived environmental information includes the position, shape, size, and other characteristics of obstacles.

(2) Path Planning: Based on the UAV's current position, target position, and environmental perception data, the best path for the UAV is calculated.

(3) Obstacle Detection: Based on the obstacle information, obstacles in front of the UAV are detected, and the UAV's movement trajectory is adusted.

(4) Path Replanning: If the UAV's path is blocked by obstacles, the best path for the UAV is recalculated based on new environmental perception data.

(5) Control Decision: The results of path planning and obstacle detection are combined to formulate the UAV's motion control strategy to achieve dynamic obstacle avoidance.

In the process of this function, the deep SARSA algorithm also needs to introduce reward and punishment mechanisms to encourage the UAV to choose the correct motion control strategy and avoid choosing the wrong strategy.

3.2.2 Deep Q-Learning Network Algorithm (DQN)

Using deep learning, the Q-learning system DQN learns new questions. DQN can handle close to the edge state spaces and continuous behavior spaces and neural networks are used to calculate the Q-value parameters [10]. DQN also introduces goal-directed networking and experience recall approaches to balance the randomness and continuity of samples, which increases learning efficiency and stability.DQN has been widely applied in robot navigation and control fields and can be used in autonomous obstacle avoidance navigation systems based on deep neural networks.

Using a deep neural network to estimate the Q-value parameters, which stands for the expected return for doing an action in a specific state, is the fundamental idea behind the DQN technique. The Q-value for each potential action is generated by the deep neural network's algorithm using the current situation as its input. The best course of action is then determined by the action with the greatest Q-value.

To address the instability and divergence issues caused by correlated samples in Q-learning, DQN introduces experience replay.The agent saves the experience tuple (state, action, reward, future state) during training into a replay storage device. To update the Q-value function, the agent randomly selects a batch of experience tuples from the replay buffer cache. By using this method, the training process is more stable and the correlation between the samples is decreased.

The target network approach is also introduced by DQN to further improve the reliability of the training process. To produce the target Q-value for the update, a copy of the Q-value network called the goal network is employed. The training process is stabilized by occasionally updating the goal network with the weights from in the Q-value network, which lessens the impact of the current iteration on the target Q-values. Deep neural network-based autonomous obstacle avoidance navigation systems may be employed using the DQN algorithm, which has been extensively utilized in the area of robot navigation and control. Here are the basic of the algorithm.

$$Y_t^{DQN} \equiv R_{t+1} + \lambda \max_a Q(S_{t+1}, a; \theta_t^-) \tag{2}$$

(1) Model the driverless routing and obstacle avoidance issue first using a Markov Decision Process (MDP) [10]. The agent in this MDP is surrounded by a state vector, an operation space, and a target value.The agent's target is to move in the environment and take a series of actions to obtain maximum cumulative reward. In this problem, the agent needs to reach the destination without colliding with obstacles.

(2) Core algorithm of DQN: DQN is a deep reinforcement learning algorithm that uses a neural network to decide the cumulative reward (Q-value) the agent can obtain when taking a certain action. The Q-values for each action are produced by the neural network from the input of the current condition in the state space. DQN uses experience replay technique to train the neural network, which means it randomly selects a small batch of data from previous experience instead of using the latest data every time. This helps reduce correlation during training, improving training efficiency and stability.

(3) Algorithm flow: Specifically, the algorithm flow of the autonomous obstacle avoidance navigation system based on DQN is as follows:

a. The state space, behavior space, and neural network are initialized.

b. At each time step, the agent chooses a behavior from as it stands at the moment. The behavior selection is based on the Q-value estimate of the neural network and can use a greedy or ε-greedy policy.

c. The agent performs the selected behavior and observes the surroundings' feedback, including the next state and reward [11].

d. The new state and reward are stored in the experience replay buffer.

e. A small batch of data is randomly selected from the experience practice buffer for playback to update the Q-value estimate of the neural network.

f. Steps (2) to (5) are repeated until the stopping condition is met.

(4) Implementation details: To improve training stability when implementing the DQN algorithm, a target network can be used. This second neural network shares the consistent basic architecture with the primary neural network, but it does not update its parameters as frequently. The target value is less unstable during training because the goal network is employed to determine the goal Q-value.

(5) Moreover, a Convolutional Neural Network (CNN) can be utilized to process the surroundings' visual input extraction of features is required and decrease the dimensionality of the state space, which will speed up practice. Both at once, the hierarchical DQN (h-DQN) algorithm can be used to divide the agent into multiple levels, each responsible for different tasks and decisions, to increase the efficiency and precision.

(6) In the autonomous obstacle avoidance navigation system, some special implementation details need to be considered. Sensors are required to obtain obstacle information in the environment and convert it into data in the state space. In addition, an appropriate reward function needs to be set to encourage the agent to stay away from obstacles and reach the destination as quickly as possible.

In summary, the autonomous obstacle avoidance navigation system based on DQN is a strengthening learning algorithm that can make agents navigate and avoid obstacles in the environment autonomously. The core of this algorithm is to use a neural network to estimate Q-values and train it using experience replay technique. During implementation, some details need to be considered, such as using a target network, CNN, and h-DQN algorithm.

3.2.3 Double Q-Learning Algorithm

It is an improvement on the Q-learning algorithm that addresses the problem of overestimation by introducing two Q-value parameters, one for behavior choosing and one for behavior evaluation. Double Q-learning algorithm divides the actions taken at each state into two sets, one for behavior choosing and one for behavior evaluation, and alternates between using the two Q-value functions to avoid overestimation [12]. While Double Q-learning algorithm is very effective in handling overestimation, it also increases computational complexity. The update formula for Double Q-learning algorithm is as follows:

$$Y_t^{DoubleQ} \equiv R_{t+1} + \beta Q(S_{t+1}, \arg\max_a Q(S_{t+1}, a; \theta_t); \theta_t^{'}) \tag{3}$$

This research proposes an obstacle avoidance algorithm for autonomous vehicles based on Double Deep Q-Learning (DDQN) and Faster R-CNN [13]. The proposed algorithm utilizes the DDQN algorithm to predict optimal action strategies while employing the Faster R-CNN algorithm to observe and recognize obstacles surrounding the vehicle and take appropriate control measures to avoid them.

DDQN is an improved version of DQN that aims to solve the overestimation problem present in DQN. DDQN uses two neural networks, as follows the target network and the local network, to estimate Q-values. The target network's parameters are updated at certain time intervals to avoid significant changes in the local network during the update process, thus reducing the occurrence of overestimation problems.

Faster R-CNN is a target detection algorithm based on CNN used to observe objects in images and label their positions. Faster R-CNN first uses CNN to further picture features and then uses the Region Proposal Network (RPN) to generate region proposals. Subsequently, each region proposal undergoes ROI pooling to transform it into a fixed-size feature map and is entry into the stratum that is entirely connected for classification and regression to obtain the object's class and position information. The detected obstacle position information is then passed to the DDQN algorithm to predict the optimal

action strategy, with the obstacle position information being input as part of the state. If the predicted action by the DDQN algorithm would result in collision with an obstacle, the vehicle will avoid the obstacle.

Faster R-CNN detects and recognizes obstacles surrounding the vehicle, providing relevant information to the DDQN algorithm [14]. Specifically, the Faster R-CNN algorithm processes the images around the vehicle, extracts features, and generates region proposals. For each region proposal, the Faster R-CNN algorithm outputs whether an obstacle exists in the region and provides its position information. This position information is then passed to the DDQN algorithm to help the vehicle avoid obstacles. Finally, the vehicle takes appropriate action, such as adjusting speed or steering, considering the results of the DDQN algorithm to avoid the detected obstacles. The autonomous vehicle's obstacle avoidance behavior is achieved by continuously alternating between the DDQN and Faster R-CNN algorithms.

In summary, this algorithm combines the DDQN and Faster R-CNN algorithms to achieve autonomous car obstacle avoidance behavior in a dynamic surroundings. These algorithms work together to help the vehicle avoid collisions with obstacles, thus achieving safe and reliable navigation and obstacle avoidance for autonomous vehicles.

4. Limitations of Q-Learning Algorithm in Dynamic Obstacle Avoidance

In dynamic obstacle avoidance problems, agents need to avoid obstacles quickly and accurately. However, traditional Q-learning algorithms have some limitations, as follows:

The first part is the state and behavior spaces are too large. In dynamic obstacle avoidance problems, factors such as obstacle position and velocity affect the agent's decision-making. Therefore, the state and behavior spaces are incredibly huge, making it difficult for Q-learning algorithms to handle this situation effectively. The second part is the reward function is difficult to design. In dynamic obstacle avoidance problems, agents need to avoid obstacles as quickly as possible, but designing a reward function is challenging. If the reward function is not well-designed, it may cause the agent to ignore long-term returns while pursuing short-term returns. The third component is the importance of striking an equilibrium between discovery and extraction. The harmony between discovery and extraction is particularly crucial in dynamic obstacle avoidance issues. If the agent is too conservative, it may miss opportunities to avoid obstacles, while if it is too adventurous, it may lead to collision accidents.

5. Conclusion

This article has highlighted the limitations of Q-learning algorithm in dynamic obstacle avoidance problems and discussed several methods for improvement. Its experimental results have demonstrated that these methods can significantly enhance the agent's performance. Furthermore, there is a need for further research to optimize and apply these methods to address more complex dynamic obstacle avoidance problems and accelerate the progress of intelligent agent technology. A promising direction could be to combine various algorithms such as deep reinforcement learning, evolutionary algorithms, and genetic algorithms to increase learning efficiency and accuracy of the agent. Additionally, the application of these improvement methods can be extended to various domains, such as autonomous driving, robot navigation, virtual gaming, among others, to promote the use of intelligent agent technology in real-life scenarios.

It is worth noting that the improvement methods discussed in this article are just a few of the many potential solutions available, and there are still many other approaches to explore. Future research can delve deeper into dynamic obstacle avoidance problems and continually propose novel solutions to advance the development of intelligent agent technology. Overall, this research area holds great potential for shaping the future of autonomous systems and enhancing their capabilities to interact seamlessly with the environment.

References

[1] Xinli Xu, Peng Cai, Zahoor Ahmed, Vidya Sagar Yellapu, Weidong Zhang, Path planning and dynamic collision avoidance algorithm under COLREGs via deep reinforcement learning, Neurocomputing, Volume 468, 2022, Pages 181-197.

[2] Guo D, Wang J, Zhao J B, et al. A vehicle path planning method based on a dynamic traffic network that considers fuel consumption and emissions [J]. Science of the Total Environment, 2019, 663: 935-943.

[3] Liu Y, Ma J, Zang S, et al. Dynamic path planning of mobile robot based on improved ant colony optimization algorithm [C] // Proceedings of the 2019 8th international conference on networks, communication and computing. 2019: 248-252.

[4] Monekosso N, Remagnino P. The analysis and performance evaluation of the pheromone-Q-learning algorithm [J]. Expert Systems, 2004, 21(2): 80-91.

[5] Y. Zhang, Y. Hu, X. Hu and B. Xing, "Path Planning for Mobile Robot Based on RGB-D SLAM and Pedestrian Trajectory Prediction," 2020 4th Annual International Conference on Data Science and Business Analytics (ICDSBA), Changsha, China, 2020, pp. 341-346.

[6] Attalla S A, Seddik K G, El-Sherif A A, et al. A Reinforcement Learning Approach to ARQ Feedback-based Multiple Access for Cognitive Radio Networks [C] // 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC). IEEE, 2021: 1-5.

[7] Xiao Y, Liu J, Wu J, et al. Leveraging deep reinforcement learning for traffic engineering: A survey [J]. IEEE Communications Surveys & Tutorials, 2021, 23(4): 2064-2097.

[8] Luo, W., Tang, Q., Fu, C., Eberhard, P. (2018). Deep-Sarsa Based Multi-UAV Path Planning and Obstacle Avoidance in a Dynamic Environment. In: Tan, Y., Shi, Y., Tang, Q. (eds) Advances in Swarm Intelligence. ICSI 2018. Lecture Notes in Computer Science, vol 10942. Springer, Cham.

[9] Sathyaraj, B.M., Jain, L.C., Finn, A., Drake, S.: Multiple UAVs path planning algorithms: a comparative study. Fuzzy Optim. Decis. Mak. 7(3), 257–267 (2008).

[10] Kim M S, Kim J S, Choi M S, et al. Adaptive Discount Factor for Deep Reinforcement Learning in Continuing Tasks with Uncertainty [J]. Sensors, 2022, 22(19): 7266.

[11] AlMahamid F, Grolinger K. Autonomous Unmanned Aerial Vehicle Navigation using Reinforcement Learning: A Systematic Review [J]. Engineering Applications of Artificial Intelligence, 2022, 115: 105321.

[12] Van Hasselt, H.; Guez, A.; Silver, D. Deep reinforcement learning with double Q-learning. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 2 March 2016.

[13] Bin Issa, R.; Das, M.; Rahman, M.S.; Barua, M.; Rhaman, M.K.; Ripon, K.S.N.; Alam, M.G.R. Double Deep Q-Learning and Faster R-CNN-Based Autonomous Vehicle Navigation and Obstacle Avoidance in Dynamic Environment. Sensors 2021, 21, 1468.

[14] Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. 2016, 39, 1137–1149.