Study of Learning based algorithms for UAV Motion Planning

by:
Manthan Patel, Jigme Tsering, Min Woo (David) Kong

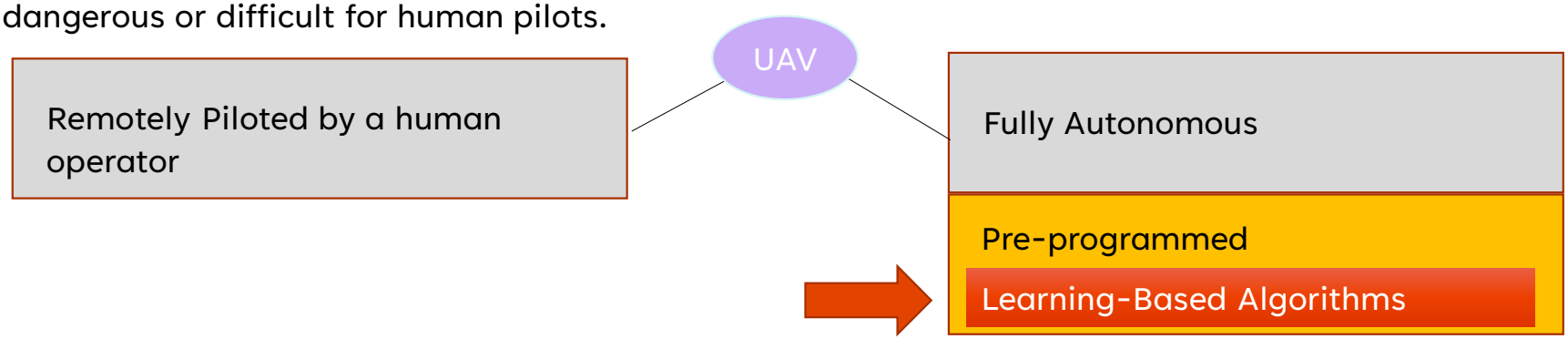
April 19, 2023

Contents

- Introduction
- Problem definition and Algorithms
- Experiment and Results
- Conclusion

Introduction

- UAV: Unmanned Aerial Vehicles that is a type of an aircraft used in a wide range of applications from surveillance and inspection to delivery and transportation as well as circumstances where they are too dangerous or difficult for human pilots.



- IMPORTANT to develop an effective motion planning technique to maximize a successful performance

- Various reinforcement learning – Proximal Policy Optimization (PPO), Deep Q-Network (DQN) , Advantage Actor Critic (A2C), and Deep Deterministic Policy Gradient (DDPG) were implemented using motion planning on Microsoft AirSim simulator software to compare their different performances.

Problem definition and Algorithms

- The goal is to use learning-based algorithm to enable quadrotor to be able to adapt to changing conditions in real-time and make decisions based on data from their sensors, such as cameras and LIDAR, avoid collisions with obstacles and reach the destination.
- Four deep reinforcement learning algorithms are selected PPO, DQN, A2C and DDPG for this task.
- These algorithms are used to train the system to pass through walls with holes in virtual environment.

Formulation of Reinforcement Learning

In deep reinforcement learning, the interactions of the robot with the environment are expressed with a triplet of state, action and reward (s,a,r)

a. State : State is obtained directly from the camera images in the form of H X W X C
b. Action space : The action of the quadrotor at time t is represented by its speed along y and z axis.
c. Reward function: The reward function consists of two terms .The first term is based on Euclidean distance between quad and gate on the wall and the second term is determined by conditions such as collisions with obstacles and visibility of the gates.

$$r_t = r_t^d + r_t^c$$

$$r_t^d = 30 \times e^{-\|p_{a,t} - p_{h,t}\|_2}$$

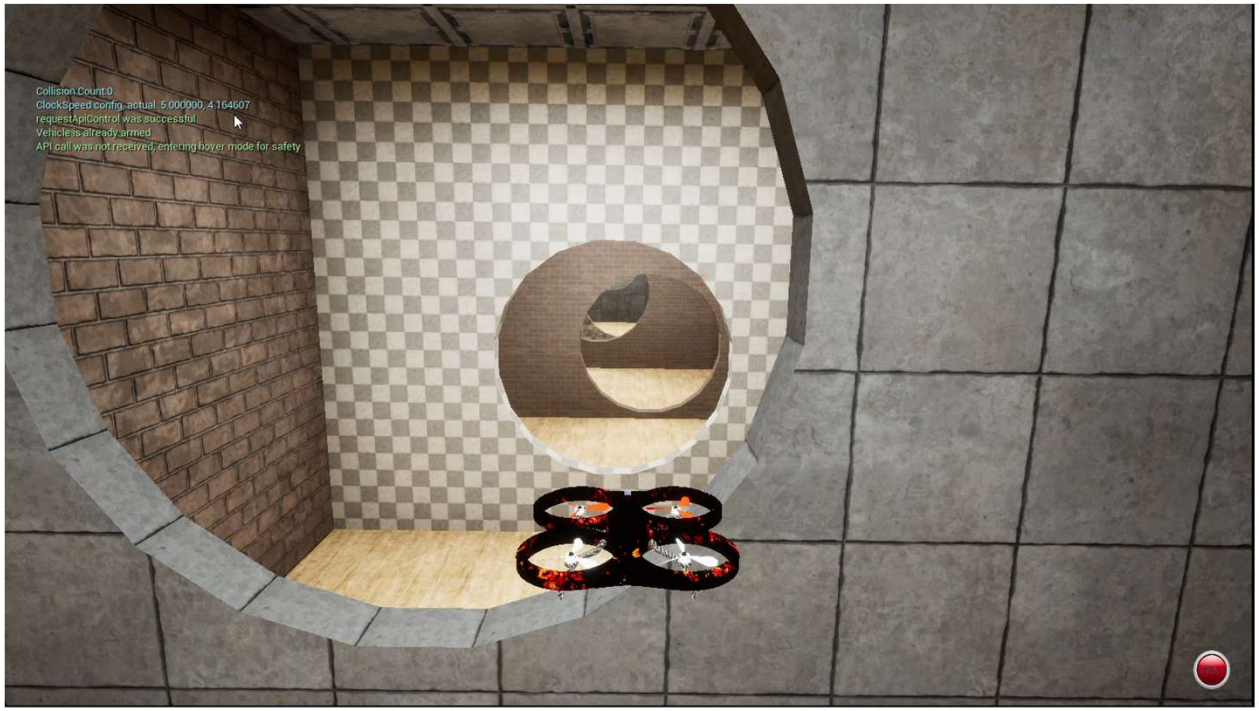
$$r_t^c = \begin{cases} -100, & C_{coll,t} \\ -100, & M_{coll,t} \\ 0, & in\,other\,cases \end{cases}$$

Implementation

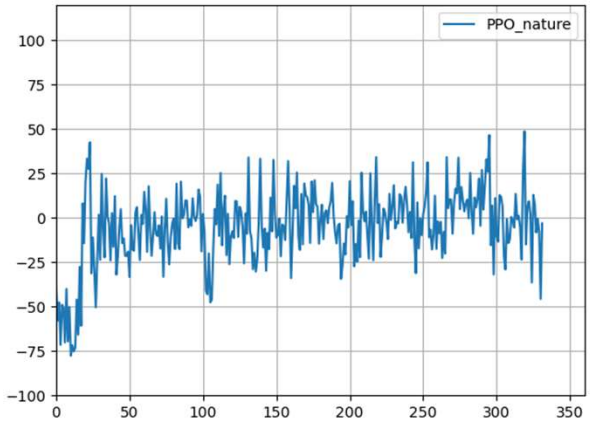
Proximal Policy Optimization (PPO)	Deep Q-Network (DQN)	Advantage Actor Critic (A2C)	Deep Deterministic Policy Gradient (DDPG)
Policy-based algorithm that optimizes the policy function to achieve better performance.	Model-free, off-policy algorithm that uses deep neural networks to approximate the Q-function, which estimates the expected reward.	Policy-based algorithm that learns both a value function and a policy function.	Model-free, off-policy algorithm that uses deep neural networks to estimate both the Q-value and the policy function.

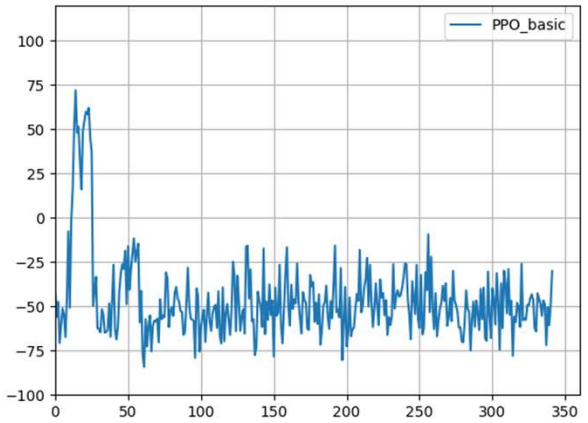
Environment Setup

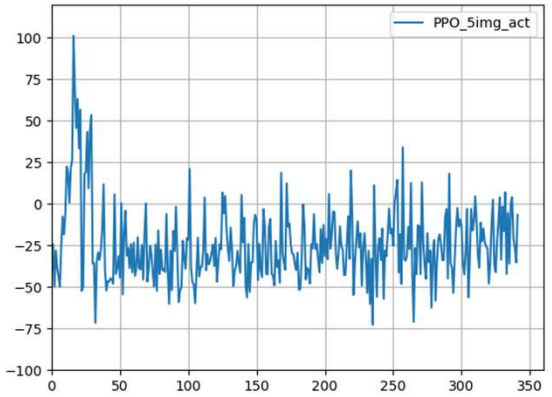
- **Simulation Setup:**
  - Microsoft AirSim API
- **Action Space**
  - [-0.6 m/s, 0.6 m/s]
  - Continuous and Discrete
- **Reward Function**
  - Collision, Gate Visibility & Distance from Goal
- **CNN Architecture**

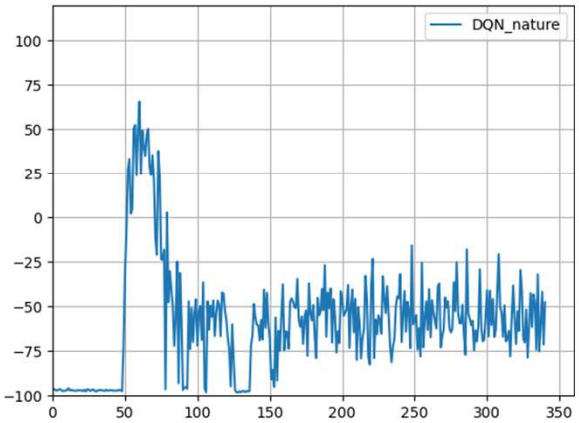


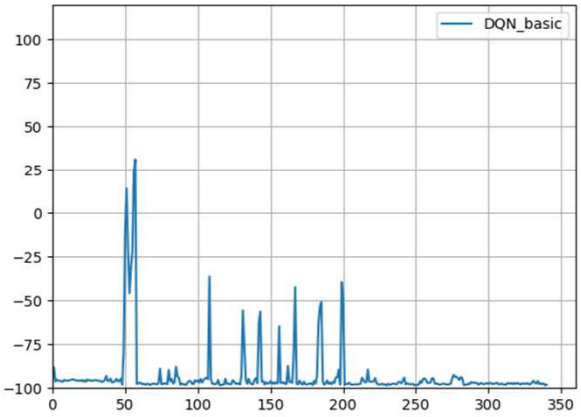
Experiment and Results



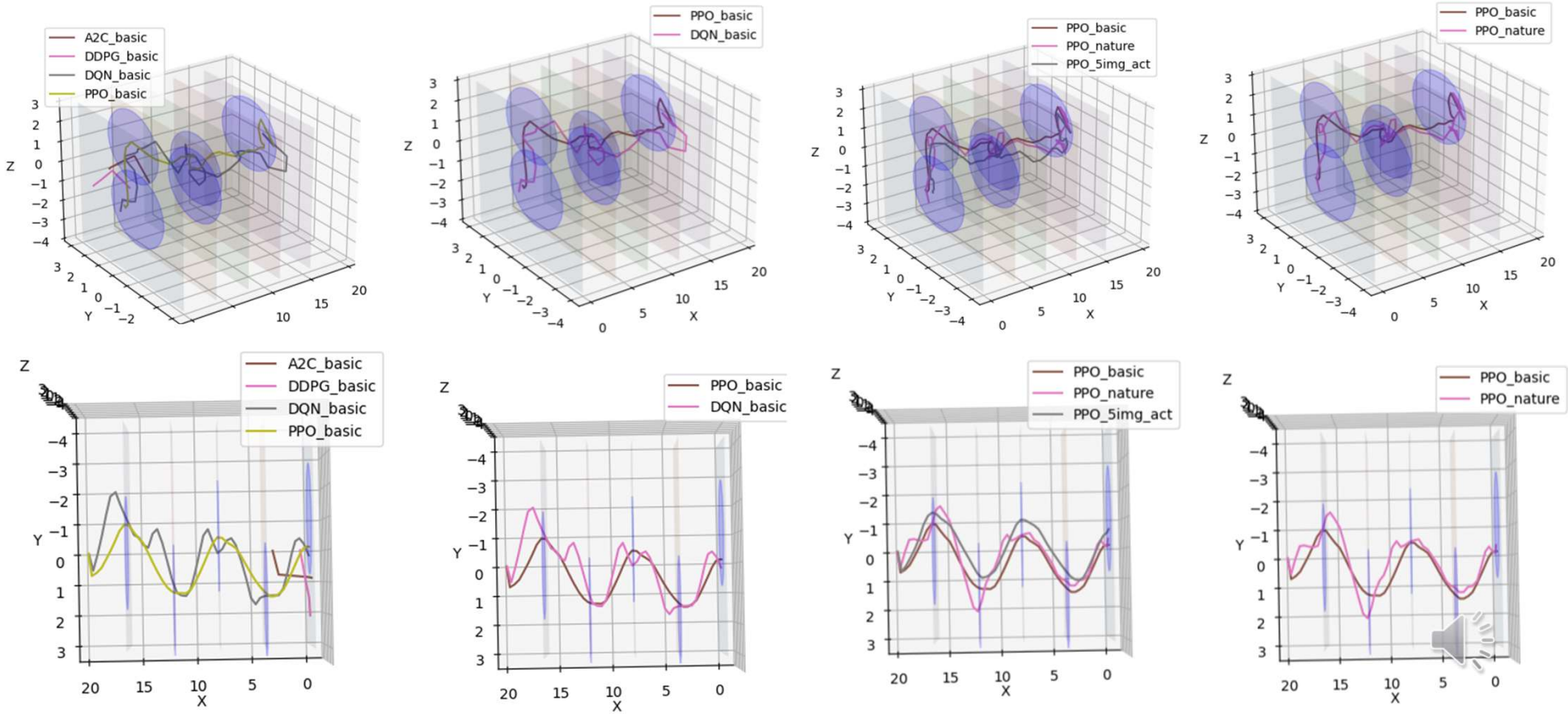








Experiment and Results



Experiment and Results

Model	Flight Distance (m)	Success Rate (Over 100 test episodes)
PPO with NatureNet	22.90 m	69 %
PPO with BasicNet	20.30 m	89 %
DDPG with NatureNet	0 m	0 %
DDPG with BasicNet	0 m	0 %
DQN with NatureNet	20.11 m	89 %
DQN with BasicNet	21.85 m	84 %
A2C with NatureNet	0 m	0 %
A2C with BasicNet	0 m	0 %
PPO with BasicNet, 5 image stack, Updated reward function, Different Action Space	20.07 m	98 %

Conclusion and Future Scope

- The conclusion of the study is that the PPO and DQN algorithms performed better than the DDPG and A2C algorithms in navigating through walls. PPO with NatureNet had a higher average reward value than PPO with BasicNet, but BasicNet had some episodes with higher rewards.
- DQN had an average reward value of around -50, while PPO had an average reward value of around 0. The choice between continuous and discrete action space depends on the task at hand. Changing the observation and action space improved the performance of PPO (BasicNet).
- DQN generated a more complex and longer path, while PPO generated a smoother and shorter path.
- The modified PPO model using 5 grayscale stacked images as input, a modified action space, and reward function generated a more centralized path with fewer collisions with walls while maintaining a comparable path length.
- future research could focus on investigating the effects of different network architectures, activation functions, and hyperparameters on the performance of reinforcement learning algorithms. Additionally, evaluating the performance of these algorithms on more complex environments can help researchers gain a better understanding of their capabilities and limitations.

Thank you!