UAV Obsatcle Avoidance Using Reinforcement Learning

Reinforcement Learning - CS3009

Jan – May 2025

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, DESIGN AND MANUFACTURING, KANCHEEPURAM

- Thakur Sai Madhan Gopal(CS22B2023)
- Pandiri Veeresh Kumar(CS22B2026)
- Dampetla Harsha Vardhan(CS22B2024)

Introduction

- Importance of UAVs: Surveillance, delivery, rescue
- Challenge: Autonomous navigation with obstacle avoidance
- Aim: Use RL to train UAVs for navigating a 3D obstacle-filled environment

Objective

- Train UAV to avoid obstacles in 3D space
- Efficiently reach targets
- Learn adaptive policies via experience
- Compare A2C vs PPO approaches

Methodology Overview

- Designed two RL-based UAV control systems in simulation.
- **A2C:** Custom-built environment using PyBullet (UAV3DEnvironment).
- **PPO:** Extended gym-pybullet-drones (ObstacleHoverAviary).
- Focus on learning from interactions using physics-based sensors and controls.

UAV3DEnvironment (A2C)

- Simulates a 3D space with randomly placed cuboid obstacles.
- **UAV:** Red sphere; **Target:** Green sphere.
- Uses 360° LIDAR-like sensing.
- State includes: position, velocity, target direction, LIDAR data.
- Continuous control in 3D (x, y, z directions).

ObstacleHoverAviary (PPO)

- Based on HoverAviary from gym-pybullet-drones.
- Adds obstacles and a fixed goal.
- Uses kinematic observations; output is RPM for drone motors.
- More realistic drone physics, but complex dynamics.

Reward Function Design (A2C)

- High reward for reaching the goal.
- Large penalties for collisions.
- Small reward for approaching the target.
- Time penalty and anti-orbiting discourages inefficient behavior.
- Encourages smooth, directed flight toward the goal.

Reward Function Design (PPO)

- Binary reward if UAV is within the goal radius.
- Minor reward for progress (based on distance).
- Penalties for being too close to obstacles or violating altitude.
- Problem: Sparse reward → weak learning signal.

A2C Algorithm Highlights

- Combines **policy** and **value** learning.
- Curriculum training:
  Phase 1: Large goal radius
  Phase 2: Medium radius
  Phase 3: Small radius
- Actor & critic networks trained together.
- Adaptive Gaussian noise for exploration.
- Momentum-based motion smoothing.

PPO Algorithm Highlights

- Uses **clipped surrogate loss** to prevent unstable updates.
- Generalized Advantage Estimation (GAE) improves learning.
- Training:
  4096 steps collected → batch updates
  Model checkpointing & evaluation
- Used vectorized environments (make_vec_env).
- Less sensitive to learning rate, but requires lots of training.

A2C Results

- **Phase 1:** Basic navigation skills learned.
- **Phase 2:** Improved obstacle avoidance.
- **Phase 3:** High precision target-reaching.
- Stable convergence with increasing environment complexity.
- Very good success rate in reaching target without collisions.

PPO Results

- Unsatisfactory performance with frequent crashes

Identified Issues:

- Insufficient training time (only 7e5 timesteps)
- Sparse reward problem (binary goal reward)
- Complex drone dynamics
- Environment differences from A2C
- Reset mechanism problems

Improving PPO Implementation

Key Improvements Needed:

- Extended training (1e6+ timesteps)
- Enhanced continuous reward function
- Curriculum learning approach
- Hyperparameter tuning
- Simpler drone dynamics initially
- Enhanced debugging and visualization

Key Learnings

- Reward function quality is **critical** in RL.
- Curriculum learning significantly boosts learning success.
- A2C showed better adaptability and stability in training.
- PPO has potential but needs more tuning and runtime.
- A2C is More Robust for Custom Simulations
- RL Algorithms Aren’t One-Size-Fits-All

Conclusion

- A2C succeeded in obstacle-avoiding UAV training.
- PPO was unstable under current settings.

Future Work

- Add computer vision sensors.
- Expand to **multi-agent** UAV navigation.
- Test in real-world or high-fidelity environments.

Individual Contributions

- **Thakur Sai Madan Gopal:**
  - Designed and implemented the A2C algorithm architecture including actor and critic network
  - structures
  - Developed the custom UAV3DEnvironment simulation environment with PyBullet integration
  - Implemented the phased curriculum learning approach with adaptive difficulty progression
  - Created the reward function engineering for the A2C implementation
  - Performed hyperparameter optimization for the A2C learning process

- **Pandiri Veeresh Kumar:**
  - Implemented the PPO algorithm using stable-baselines3 framework
  - Developed the ObstacleHoverAviary environment based on gym pybullet drones
  - Created the reward function for the PPO implementation
  - Identified limitations in the current PPO implementation and proposed improvements
  - Conducted comparative analysis between A2C and PPO performance metrics

Individual Contributions (cont.)

Damptla Harsha Vardhan(CS22B2024)

- Implemented the collision detection and handling system for the simulation environments
- Developed the LIDAR-based sensing capabilities for obstacle detection
- Created the momentum-based physics model for realistic UAV movement
- Optimized the environment reset mechanism and obstacle generation algorithms
- Authored significant portions of the project documentation and final repor

GITHUB LINK

- https://github.com/Thakursaimadan/RL-PROJECT

THANK YOU