SkyRoute: Autonomous Drone Delivery System using Approximate Q Learning

Anustup Bhaumik , Debanjan Kola , Debrup Chatterjee
Ramakrishna Mission Vivekananda Educational and Research Institute

Abstract

This report discusses a technical study of an autonomous drone delivery system trained using Approximate Q-learning with linear function approximation. We model this task as a finite Markov Decision Process (MDP) within a Gym-compatible simulation. The agent learns a navigation policy that balances path efficiency, obstacle avoidance, and successful task completion based on a defined reward function. Linear approximation helps us deal with the scalability challenges of tabular Q-learning in navigation problems that involve large state spaces. Experimental results show that the system converges to a stable and interpretable policy that can complete the delivery task, but they also highlight known stability issues with off-policy learning and function approximation.

Introduction

Problem Definition

Autonomous drone delivery involves making decisions sequentially while considering constraints like obstacles, boundaries, and limited resources. The goal is to create a policy that directs the drone from a starting point to a delivery target, minimizing time and avoiding collisions.

We model this problem as a finite Markov Decision Process (MDP). At each discrete time step t, the agent observes a state s_t, chooses an action a_t, receives a reward R_{t+1}, and moves to a new state s_{t+1}. The aim is to maximize the expected discounted return.

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}, \tag{1}$$

where $\gamma \in (0, 1)$ is the discount factor.

Relevance in Reinforcement Learning

Reinforcement Learning (RL) offers a suitable framework for autonomous navigation, allowing an agent to learn optimal behavior through interactions with the environment without needing a model of transition dynamics. However, classical tabular Q-learning is not practical when the state space is large or continuous.

To overcome this limitation, we use Approximate Q-learning with linear function approximation. This method allows us to generalize across states using a compact set of designed features, providing a more efficient alternative to deep reinforcement learning for medium-scale navigation tasks.

Project Objectives

The objectives of this study are:

- To model an autonomous drone delivery task as a Gym-compatible MDP.
- To implement Approximate Q-learning using a linear action–value function.
- To select stable hyperparameters for learning and exploration.
- To evaluate the learned policy and analyze convergence behavior.

Problem Formulation

Environment Domain

The environment is a discrete, grid-based navigation space based on a map layout. While simplified from real-world drone dynamics, it includes important constraints such as obstacles, boundaries, and final goal conditions. Episodes end with either successful delivery or failure.

State and Action Spaces

The raw state includes the drone's position on the grid. For value approximation, the agent uses a feature representation

$$\mathbf{f}(s, a) \in \mathbb{R}^4,$$

which encodes task-relevant properties of the state–action pair.

The action space is discrete and finite, consisting of basic movement primitives corresponding to directional navigation decisions.

Reward Structure

The reward function is designed to encourage efficient and safe navigation:

- A large positive reward for successful delivery.

- A small negative step cost to penalize long paths.
- Large negative penalties for collisions or invalid actions.

Table 1 summarizes the reward structure.

Reward Type	Value (Approx.)	Purpose
Success Reward	$+100$	Terminal objective
Step Penalty	-1	Path efficiency
Failure Penalty	-100 to -300	Safety enforcement

Table 1: Reward structure used in the drone navigation environment.

Methodology

Approximate Q-Learning

Approximate Q-learning is an off-policy temporal-difference control algorithm. The action–value function is approximated as

$$\hat{Q}(s, a; \mathbf{w}) = \mathbf{w}^T \mathbf{f}(s, a), \tag{2}$$

where $\mathbf{w} \in \mathbb{R}^4$ is the weight vector.

Action selection follows an ϵ-greedy policy, where the agent explores with probability ϵ and exploits the current estimate otherwise.

Temporal Difference Update

The temporal-difference (TD) error is computed as

$$\delta_t = R_{t+1} + \gamma \max_{a'} \hat{Q}(s_{t+1}, a') - \hat{Q}(s_t, a_t). \tag{3}$$

The weights are updated using stochastic gradient descent:

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha \delta_t \mathbf{f}(s_t, a_t), \tag{4}$$

where α is the learning rate.

Feature Representation

The linear approximation relies on four hand-engineered features hypothesized to encode:

1. Progress toward the goal.
2. Proximity to obstacles or boundaries.
3. Time or movement cost.
4. Terminal success indication.

The magnitudes of the learned weights indicate that penalty-related features dominate the policy, resulting in strong risk-averse behavior.

Experimental Setup

Training was conducted over approximately 500 episodes using a decaying ϵ-greedy schedule, with ϵ decreasing from 0.99 to 0.01. A high discount factor ($\gamma = 0.99$) was used to propagate the delayed success reward back through the action sequence.

Results

This section presents a qualitative and behavioral evaluation of the trained agent under varying environmental conditions. In addition to the final successful delivery outcome, intermediate failure cases are analyzed to highlight the limitations of Approximate Q-learning with linear function approximation in dynamic, stochastic environments involving wind disturbances and moving obstacles.

All results are obtained using the learned policy under low exploration ($\epsilon \approx 0.01$), ensuring that the observed behaviors reflect exploitation of the learned action–value function rather than random exploration.

Effect of Wind Intensity on Navigation Behavior

The wind field introduces a stochastic external force that perturbs the drone's motion. The agent was evaluated under three representative wind regimes: moderate wind, high wind, and stabilized wind.

Moderate Wind: Loss of Stability Under moderate wind intensity, the drone exhibits partial instability. While the agent attempts to move toward the goal, wind-induced drift causes repeated corrective actions, resulting in inefficient zig-zag trajectories. The policy remains goal-directed but struggles to maintain smooth motion.

High Wind: Stagnation and Battery Depletion At higher wind intensities, the learned policy fails to make forward progress. The wind force repeatedly counteracts the agent's selected actions, causing the drone to remain trapped in a small region of the state space. This results in visible shaking behavior, characterized by rapid back-and-forth movements.

Due to the per-step battery consumption, the drone eventually exhausts its battery while remaining stationary, leading to episode termination with a large negative reward.

Figure 1 demonstrates this failure mode, where the drone becomes stuck and exhibits continuous oscillation.

These results indicate that the linear approximation lacks the expressive capacity to represent robust corrective strategies under strong external disturbances.

Interaction with Dynamic Obstacles (Birds)

In addition to wind, the environment contains dynamic obstacles in the form of birds with independent motion patterns. Although the learned policy strongly penalizes proximity to obstacles, it does not explicitly model future bird trajectories.

As a result, in some episodes the drone successfully avoids birds, while in others it collides with a bird due to delayed or inaccurate value estimation of dynamic risk.

Figure 2 shows a representative collision event, where the drone intersects with a bird's trajectory, resulting in immediate episode termination.

This behavior highlights a limitation of the feature-based linear Q-function, which lacks explicit temporal prediction of moving obstacles.

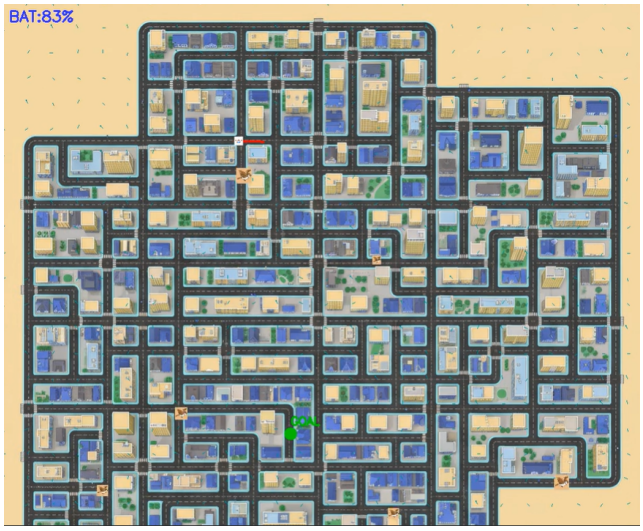

Figure 1: Failure case under high wind intensity. The drone becomes trapped in a local region, oscillating without progress while battery levels steadily decrease.

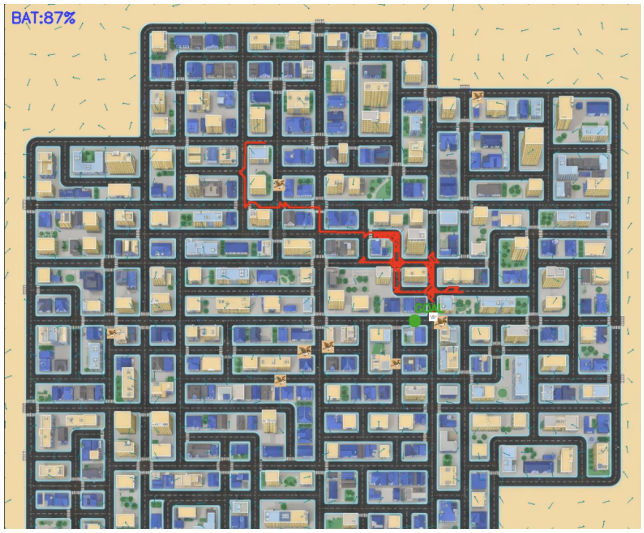

Figure 2: Collision with a dynamic obstacle (bird). The agent fails to anticipate the bird's future position, leading to a catastrophic termination.

Successful Delivery Case

Despite the observed failure modes, the trained agent is capable of successfully completing the delivery task under stabilized wind conditions and favorable obstacle configurations.

Figure 3 presents the final successful trajectory, where the drone follows an efficient, collision-free path and reaches the goal state within the battery constraints.

This outcome confirms that Approximate Q-learning with linear function approximation can learn a functional navigation policy, provided that environmental stochasticity remains within manageable limits.

Summary of Observed Behaviors

Across all evaluated scenarios, the learned policy exhibits the following characteristics:

- Strong aversion to collisions and boundary violations.
- Sensitivity to wind-induced dynamics, particularly at high intensities.
- Limited ability to anticipate dynamic obstacle motion.
- Reliable goal-reaching behavior under stabilized environmental conditions.

These observations are consistent with the theoretical strengths and limitations of off-policy reinforcement learning with linear function approximation.

Discussion

The results confirm the effectiveness of Approximate Q-learning with linear function approximation for medium-scale navigation tasks. The learned policy focuses on safety and efficiency, as indicated by the high negative penalty weights.

However, the instability observed highlights theoretical issues with combining off-policy bootstrapping and function approximation. Additionally, the linear model limits the agent's ability to capture non-linear interactions between features.

Conclusion

This work shows that Approximate Q-learning with linear function approximation can effectively tackle an autonomous drone delivery navigation problem in a simulated environment. The approach is efficient, interpretable, and capable of creating robust policies based on structured reward functions.

Future research should work on improving stability and representation challenges through algorithmic enhancements.

Future Work

Potential extensions include:

- Double Q-learning to reduce overestimation bias.
- Experience replay to improve learning stability.
- Deep Q-Networks (DQN) for automatic non-linear feature learning .

References

[Watkins and Dayan, 1992] Watkins, C. J. C. H. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3–4):279–292.

[Dietterich, 2000] Dietterich, T. G. (2000). Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research*, 13:227–303.

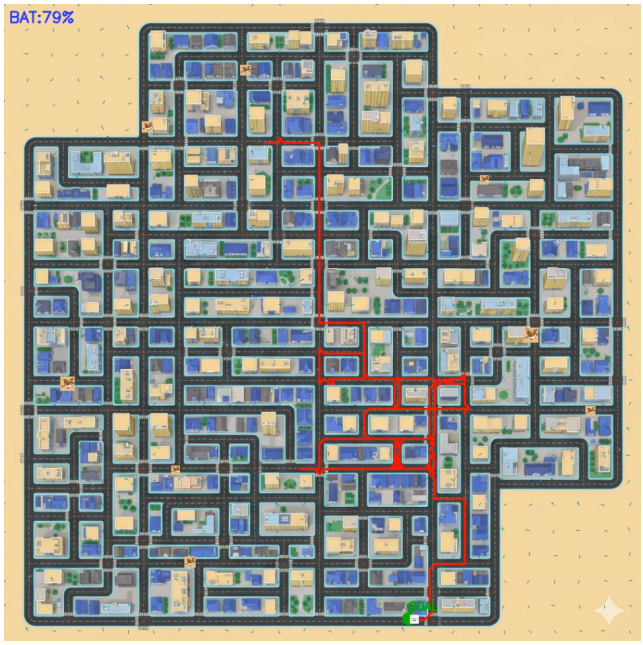

Figure 3: Successful delivery episode. The drone reaches the goal efficiently while avoiding obstacles, demonstrating the learned policy's effectiveness under moderate environmental conditions.

[Kim, 2024] Kim, C. (2024). Discrete space deep reinforcement learning algorithm based on support vector machine recursive feature elimination. *Symmetry*, 16(8):940.