# Reinforcement Learning for Air Traffic Control

Pranjal Rastogi[1], Shivangi Agarwal[2], Satvik Bajpai[3]

[1]U20220066, Plaksha University, India
[2]U20220103, Plaksha University, India
[3]U20220079, Plaksha University, India

[1]pranjal.rastogi@plaksha.edu.in; [2]satvik.bajpai@plaksha.edu.in [3]shivangi.agarwal@plaksha.edu.in;

**Abstract**

This paper presents a reinforcement learning approach for autonomous aircraft guidance in complex airspace environments. We formulate the air traffic control problem as a Markov Decision Process (MDP) with a continuous action space and implement a Proximal Policy Optimization (PPO) agent that guides aircraft to safely align with runway approach corridors. Our model incorporates realistic physics, wind effects, fuel management, and minimum vectoring altitude constraints.

**Keywords**— Reinforcement Learning, Air Traffic Control, Curriculum Learning, Proximal Policy Optimization, Flight Management Systems, Autonomous Navigation, Reward Shaping

## 1 Problem Statement and MDP Formulation

### 1.1 Problem Statement

Air Traffic Control (ATC) systems are crucial for managing the safe and efficient movement of aircraft in airspaces worldwide. The increasing volume of air traffic necessitates the development of automated decision-making systems that can assist or augment human controllers. This project addresses the challenge of guiding aircraft from arbitrary entry points to a safe landing approach corridor through a complex airspace environment containing minimum vectoring altitude (MVA) restrictions, wind influences, and fuel constraints.

We formulate this problem as an aircraft guidance task where the agent must:

1. Navigate the aircraft from any point in the airspace toward a runway to align itself for landing by reaching the Final Approach Fix (FAF)
2. Maintain safe altitude above the terrain (MVA constraints)
3. Manage fuel efficiently
4. Compensate for variable wind conditions that affect aircraft performance
5. Avoid collisions and work for complex terrains, bad weather and multiple aircraft scenarios

The aircraft dynamics model incorporates realistic physics. We have enhanced the simulator originally developed by F. Valka [1] by adding improved visualization capabilities, accurate fuel consumption modeling, and wind effects based on existing research.

## 1.2 MDP Formulation

| **Component** | **Description** |
|---|---|
| **State Space** | 10 normalized features per aircraft:<br>• Position $(x, y)$ in nautical miles<br>• Altitude $(h)$ in feet<br>• Heading $(\phi)$ in degrees<br>• Speed $(v)$ in knots<br>• Height above MVA<br>• Distance to FAF<br>• Relative angle to FAF<br>• Relative angle to runway<br>• Fuel percentage remaining |
| **Action Space** | Three continuous parameters normalized to $[-1, 1]$:<br>• Speed $(v)$: $[100, 300]$ knots<br>• Altitude $(h)$: $[0, 38000]$ feet<br>• Heading $(\phi)$: $[0, 360]$ degrees |
| **Reward Function** | Multi-component reward structure:<br>• Time penalty ($-0.5 \times$ timestep)<br>• Action smoothness penalties for rapid changes<br>• Fuel depletion penalty ($-10$)<br>• MVA violation penalty ($-50$)<br>• Airspace boundary penalty ($-100$)<br>• Success reward ($200 + \text{time\_bonus} + \text{fuel\_bonus}$)<br>• Approach position and alignment rewards<br>• Fuel efficiency bonuses<br>• Collision penalties if two planes come within 5 nautical miles of each other and are in 1000 feet of each other. |
| **Terminal States** | Episode termination conditions:<br>• Success: Aircraft enters approach corridor (reaches FAF)<br>• Failure: Aircraft descends below MVA<br>• Failure: Aircraft exits defined airspace<br>• Failure: Aircraft runs out of fuel<br>• Failure: Maximum steps ($5 \times 10^6$) reached<br>• Failure: Collission between aircrafts (for multiple aircraft scenario) |

Table 1: Markov Decision Process (MDP) formulation for the ATC guidance task

**Transition Dynamics:** The environment models realistic aircraft physics including position updates based on ground speed (affected by wind), heading changes limited by bank angle restrictions, altitude changes limited by climb/descent rates, speed adjustments limited by aircraft performance, fuel consumption based on thrust settings and flight phase, and wind effects on aircraft track versus heading (crab angle).

# 2 Methodology and Contributions

## 2.1 Methodology

We implemented a reinforcement learning system using Proximal Policy Optimization (PPO) from Stable Baselines 3 with several enhancements tailored to the ATC domain:

### 2.1.1 Training Scenarios

We trained and evaluated our models on realistic scenarios based on actual flight data:

- Primary Scenario: Based on a flight to KNEW (New Orleans Lakefront Airport) at 6 ft elevation
- Real-world Reference: Flight trajectory from ADS-B Exchange (ICAO: a36ffc) recorded on January 1, 2025
- Scale: 1 Degree Latitude = 60 nautical miles, with 1-second timesteps
- Curriculum Points (for curriculum learning): 30-100 entry points generated deterministically along the approach path

This realistic scenario incorporates actual MVA regions, allowing a direct comparison with real-world flight patterns.

### 2.1.2 Training Scenarios

- No-Go Airspace Scenario - This scenario was to simulate situations meant to be avoided under various conditions such as cyclones, storms, no-go airspaces, etc. To set up this scenario, we defined MVAs with a very large height. It is impossible for the plane to fly "through" this storm and it has to find a path around it.

- Fuel Optimization Scenario - To encourage fuel-efficient flight behavior, we initialized the airplane with a reduced fuel reserve. The goal in this scenario is for the agent to leverage favorable wind patterns to conserve fuel while still reaching the runway. The agent must learn to optimize its route to take advantage of wind-induced acceleration.

- Two-Airplane Scenario - In this multi-agent scenario, two aircraft are present, and the agent is responsible for guiding both to their Final Approach Fix (FAF) while maintaining a safe separation. Collisions or unsafe proximity between the aircraft are heavily penalized (details provided later), making coordination and spatial awareness critical for success.

- Generalized Scenario - This scenario introduces variability in MVAs, wind conditions, and entry points during training. It is designed to evaluate the agent's ability to generalize and perform effectively under realistic, dynamic conditions where it must reach the runway from any direction.

### 2.1.3 Training experiments

- Deep-Q Networks - We also experimented with training using Deep Q-Networks by discretizing the continuous action space into a finite set of classes. Specifically, we tested with discretizations of 800, 3,400, 12,500, and 48,700 action classes. However, across all configurations, the models performed poorly, even after extensive training over many episodes. The best-performing model still yielded average rewards of approximately -3000, indicating that DQN was ineffective in this setting.

- Proximal Policy Optimization - Our attempts to implement a PPO model from scratch proved to be quite challenging. The training process was unstable, and we observed very slow convergence—even in relatively simple scenarios. Despite extensive tuning and prolonged training, the model failed to demonstrate consistent improvements in performance, suggesting that either the implementation required further refinement or PPO may not be well-suited to our problem setup in its current form.

- PPO using StableBaselines3 - Using the PPO implementation from the Stable Baselines3 library yielded significantly better results compared to our custom implementation. The training process was more stable, and the model demonstrated consistent convergence across various scenarios. Additionally, this approach scaled well to more complex environments, making it a practical and reliable choice for our problem domain.

- Curriculum Learning - We adopted a curriculum learning approach to gradually increase the difficulty of the task during training. The curriculum was structured in multiple stages, with each stage defined by the aircraft's starting distance from the Final Approach Fix (FAF). In the initial stages, the aircraft starts relatively close to the FAF, and in subsequent stages, the distance is incrementally increased. To maintain consistency and reduce variance during training, the initial heading toward the FAF in each stage was constrained to lie within a narrow ±5° range. The agent advanced to the next stage only after achieving a success rate of 95% over a sliding window of episodes, ensuring reliable performance before increasing complexity. This staged progression allowed the agent to first master simpler navigation tasks before being exposed to more challenging scenarios, ultimately leading to better learning efficiency and stability.

### 2.1.4 Reward Shaping

To guide the agent towards safe and efficient landings, we employ reward shaping. The total reward $R_t$ at timestep $t$ is a sum of several components designed to incentivize desired behaviors and penalize unsafe actions.

- **Time Penalty:** A constant penalty per timestep encourages efficiency: $R_{time} = -c_{time} \cdot \Delta t$, where $\Delta t$ is the simulation timestep duration and $c_{time}$ is a small positive constant.
- **Terminal Rewards/Penalties:** Large rewards or penalties are given at episode termination:
  - *Success:* A positive reward $R_{success} = c_{succ} + c_{fuel} \cdot F_{rem\%} + c_{time\_bonus} \cdot \max(0, T_{limit} - t)$ is awarded for entering the final approach corridor, incorporating bonuses for remaining fuel percentage $F_{rem\%}$ and time efficiency (based on timestep limit $T_{limit}$ and current step $t$).
  - *Failure:* Significant negative penalties are applied for MVA violations ($R_{MVA} = -c_{MVA}$), leaving the airspace ($R_{airspace} = -c_{airspace}$), collisions ($R_{collision} = -c_{collision}$), or running out of fuel ($R_{oof} = -c_{oof}$).

- **Approach Shaping:** Dense rewards guide the aircraft towards the Final Approach Fix (FAF):
  - *Progress:* Reward for decreasing distance to FAF ($d_{FAF}$): $R_{progress} = c_{prog} \cdot \max(0, d_{FAF,t-1} - d_{FAF,t})$.
  - *Closeness:* Sigmoid-based reward increasing as $d_{FAF}$ decreases: $R_{close} = \frac{c_{close}}{1+e^{k_{close}(d_{FAF}-d_{thresh})}}$.
  - *Alignment:* When close to the runway, reward for aligning aircraft heading $\phi_{ac}$ with runway heading $\phi_{rwy}$: $R_{align} = c_{align} \cdot (1 - \frac{|\phi_{rwy}-\phi_{ac}|_{rel}}{180})$ if $d_{FAF} < d_{align\_thresh}$.
- **Behavior Penalties:**
  - *Circling:* Penalty proportional to heading change: $R_{circle} = -c_{circle} \cdot |\phi_{ac,t} - \phi_{ac,t-1}|_{rel}$.
  - *Invalid Actions:* A small penalty $R_{invalid} = -c_{invalid}$ is applied if the agent commands an action outside the aircraft's operational limits.

The constants $(c_{time}, c_{succ}, c_{fuel}, \dots)$ are present in the appendix.

### 2.2 Contributions

Our work makes the following notable contributions:

1. **Curriculum Learning for ATC**: A strategy that progressively increases task difficulty by adjusting aircraft starting positions, with automatic advancement to more challenging scenarios when consistent success is achieved.

2. **Comprehensive Environment Model**: Significant enhancements to the original simulation including realistic wind effects, fuel consumption modeling, and gradient-based minimum vectoring altitude regions.

3. **Real-world ATC Scenario**: Implementation of the KNEW scenario based on real-world airspace data, incorporating MVA regions, approach corridors, and runway geometry that mirror real ATC challenges for realistic training and evaluation.

4. **Enhanced Visualization**: Advanced visualization including colour gradient-based MVA visualization, vector representations of wind fields, realistic aircraft models, and comprehensive trajectory tracking, enabling more intuitive debugging and clearer interpretation of agent behavior.

## 3 Results

### 3.1 Training Performance

The training performance of our PPO agent improved significantly when using curriculum learning compared to standard end-to-end training.

This demonstrates that curriculum learning provides substantial advantages in mastering complex sequential tasks like aircraft guidance.

### 3.2 Total Reward Analysis

The total reward per episode across different scenarios highlights the agent's learning effectiveness and the varying challenges of the tasks.

**Fuel Optimization Scenario (Figure 1):** The learning curve shows a clear increase in total reward from negative values towards significant positive rewards (400-600) around episode 140. This indicates successful learning and optimization of the flight path while managing fuel.

**Real World Scenario (Figure 2):** Training in the Real World scenario shows rapid convergence, with total reward quickly rising from large negative values to stabilize at a high positive level (400-500) within the first 30-40 episodes. This demonstrates efficient mastery of navigation and constraints in this environment.

**Two Plane Scenario (Figure 3):** The Two Plane scenario proves significantly more challenging. While initial episodes show some reward increase from highly negative values (-25000 to -10000), the reward plateaus around -10000 to -5000. The persistently high negative rewards suggest the agent struggles to consistently avoid collisions or maintain separation, incurring large penalties despite other learned behaviors. Rewards are visible in Figure 7.

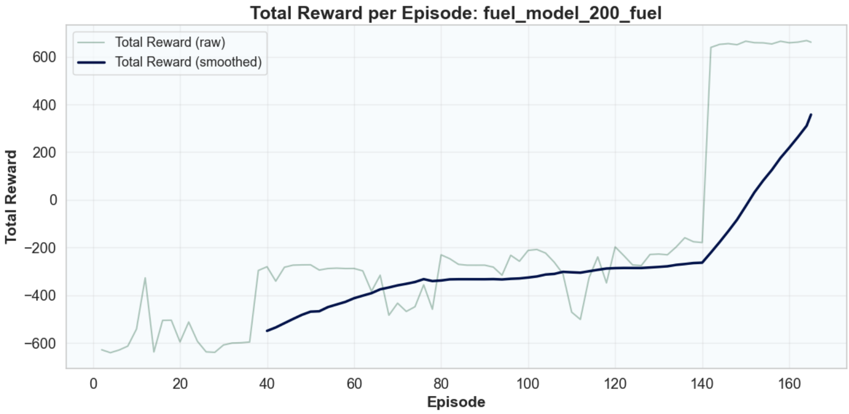

Figure 1: Reward Accumulation for Fuel Optimization Scenario

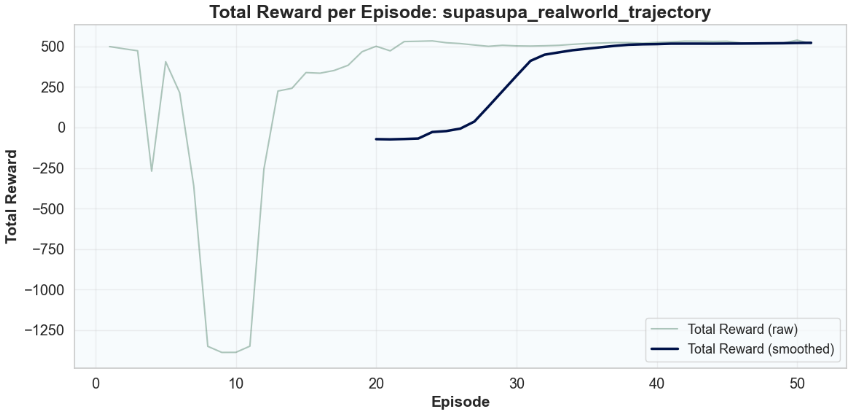

Figure 2: Reward Accumulation for Real World Scenario

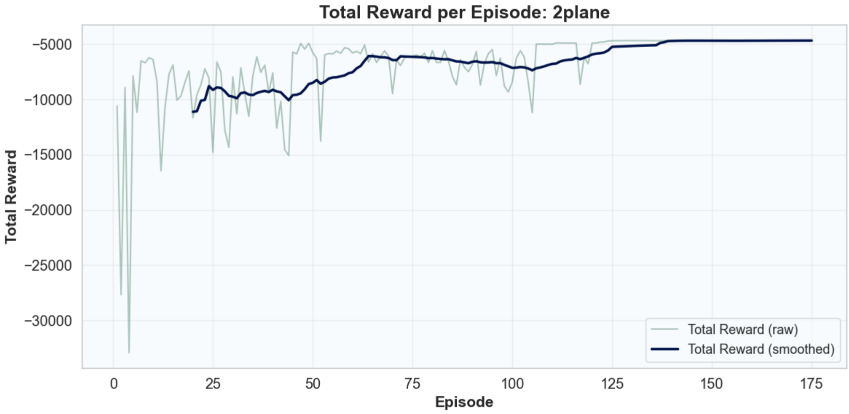

Figure 3: Reward Accumulation for Two Plane Scenario

### 3.3 Trajectory Comparison with Real-World Flight Data

To validate our approach against real-world operations, we compared trajectories generated by our RL agent with actual flight data from ADS-B Exchange for the KNEW scenario. For a fair comparison, we cropped the final portions of both trajectories (25 points from our trajectory and 64 points from the real trajectory), as the real aircraft executed a large loop prior to landing that was outside our simulation scope.

| Metric | Value |
|---|---|
| Trajectory Match | 90.59% |
| Fuel Consumption | 36.94 vs. 43.79 units |
| Path Length | 46.18 vs. 54.73 nm |
| Avg. Pointwise Distance | 4.75 nm |
| Hausdorff Distance | 4.94 nm |
| Mean Curvature | 0.0004 vs. 0.084 |

Table 2: Comparison between RL agent trajectory and real-world flight data

The high trajectory match percentage (90.59%) indicates strong alignment between our agent's flight path and the real-world trajectory. The Hausdorff distance (4.94 nm) measures the maximum deviation between the two paths, confirming good overall spatial correspondence. Our analysis revealed several notable differences:

- **Fuel Efficiency**: The agent achieved a 15.63% reduction in fuel consumption (36.94 vs. 43.79 units) while maintaining identical fuel efficiency per unit distance (0.80).
- **Path Smoothness**: Significantly lower mean curvature (0.0004 vs. 0.084) indicates smoother turns and more gradual heading changes than the real flight.
- **Directness**: Our agent's trajectory was more direct (46.18 vs. 54.73 nm) while still maintaining realistic flight dynamics and MVA constraints.

Table 3 illustrates both trajectories overlaid on the same airspace. The agent's path exhibits more consistent approach angles with fewer heading variations compared to the real flight, which shows more adjustments possibly due to ATC instructions or weather conditions not modeled in our simulation.

These results demonstrate that our reinforcement learning approach produces realistic and efficient flight paths that closely match actual operations while significantly improving fuel efficiency. The high trajectory match percentage and smoother flight characteristics indicate that the agent has learned generalizable navigation principles that align with and potentially improve upon real-world piloting practices.

### 3.4 Trajectory Comparison for Fuel Optimization Scenario

To validate our approach against real-world operations, we compared trajectories generated by our RL agent with actual flight data from ADS-B Exchange for the KNEW scenario. For a fair comparison, we cropped the final portions of both trajectories, as the real aircraft executed a large loop prior to landing that was outside our simulation scope.

The Hausdorff distance measures the maximum deviation between two paths, confirming good spatial correspondence in both scenarios. The low-fuel scenario demonstrates the agent's ability to adapt to fuel constraints while maintaining high trajectory match (92.17%). Notable observations include:

| **Metric** | **Low-Fuel Scenario** |
|---|---|
| Trajectory Match | 92.17% |
| Fuel Consumption (agent vs. real) | 21.70 vs. 25.80 units |
| Fuel Improvement | 15.87% |
| Path Length (agent vs. real) | 27.13 vs. 32.24 nm |
| Avg. Pointwise Distance | 2.32 nm |
| Hausdorff Distance | 3.59 nm |
| Mean Curvature (agent vs. real) | 0.037 vs. 0.042 |

Table 3: Comparison between RL agent trajectory and real-world flight data in low-fuel scenario

- **Fuel Efficiency**: The agent achieved consistent fuel reduction across both scenarios (15.63% and 15.87%), while maintaining identical fuel efficiency per unit distance (0.80).
- **Path Alignment**: The low-fuel scenario showed improved spatial alignment with the real trajectory (avg. pointwise distance of 2.32 nm vs. 4.75 nm in the standard scenario).
- **Flight Characteristics**: In the low-fuel scenario, curvature values were much closer between agent and real trajectories (0.037 vs. 0.042), indicating similar turning patterns when operating under strict fuel constraints.

Figure 8 illustrates the trajectories for both scenarios. The consistent performance across different fuel conditions demonstrates that our reinforcement learning approach produces realistic and efficient flight paths that closely match actual operations while significantly improving fuel efficiency. These results indicate that the agent has learned generalizable navigation principles that adapt to different operational constraints while maintaining alignment with real-world practices.

# References

[1] F. Valka. *Reinforcement Learning for Air Traffic Control Environment*. https://github.com/fvalka/atc-reinforcement-learning. 2020.

# A Reward Shaping Constants

- $c_{time}$: 0.05
- $c_{succ}$: 800 (base success reward)
- $c_{fuel}$: 30 (fuel bonus multiplier)
- $c_{time\_bonus}$: 0.5 (time bonus multiplier)
- $c_{MVA}$: 3000 (penalty magnitude)
- $c_{airspace}$: 3000 (penalty magnitude)
- $c_{collision}$: 3000 (penalty magnitude)
- $c_{oof}$: 3000 (penalty magnitude)
- $c_{prog}$: 0.005 (progress reward multiplier)
- $c_{close}$: 100 (closeness reward scaling factor)
- $c_{align}$: 100 (alignment reward scaling factor)
- $c_{circle}$: 0.005 (circling penalty multiplier)
- $c_{invalid}$: 0.1 (invalid action penalty)

# B Supplementary Figures

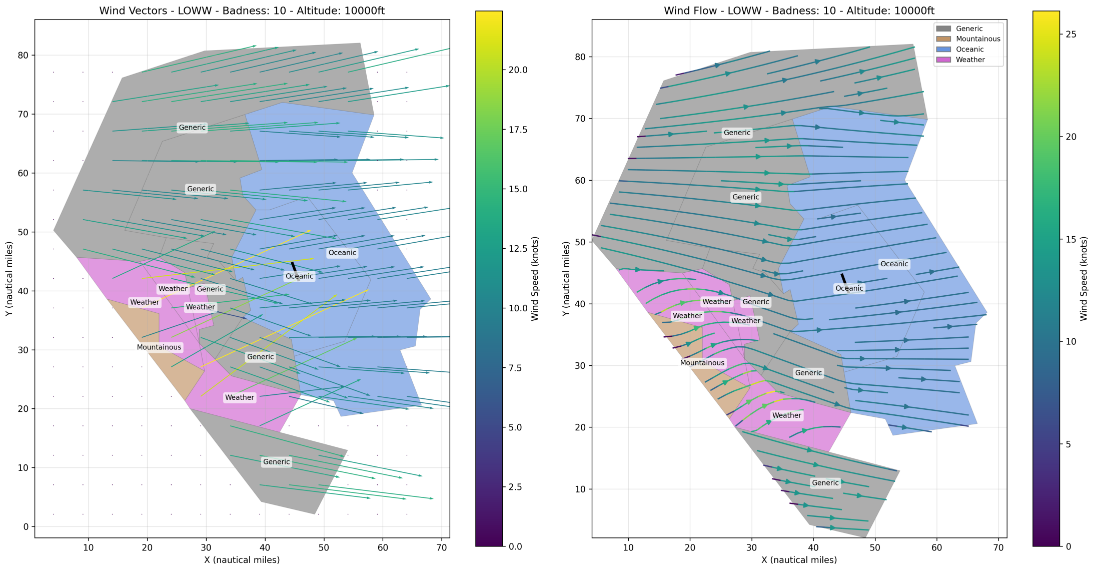

Figure 4: Wind Visualization in our environment at badness level 10. The first image represents wind vectors at each point, whereas the second image represents the overall wind flow direction in the scenario.

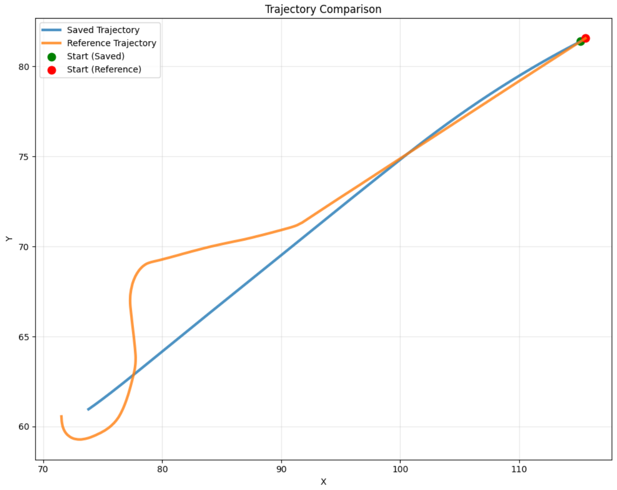

Figure 5: Trajectory comparison for real world KNEW scenario - blue trajectory is our agent, orange is the real word trajectory.

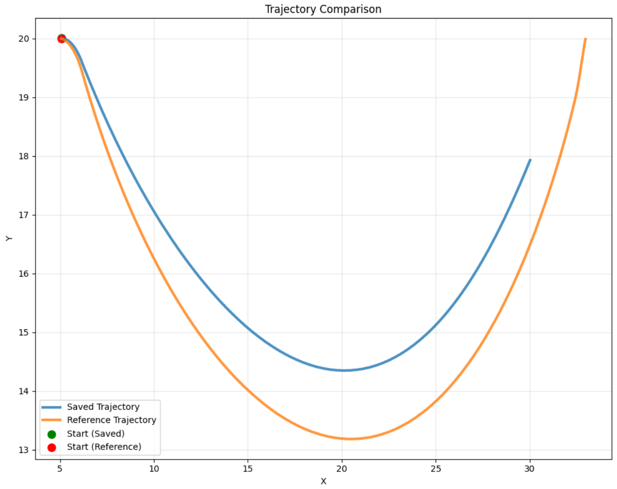

Figure 6: Trajectory comparison for fuel efficiency scenario - blue trajectory is our fuel-optimized agent, orange is non-fuel optimized agent.

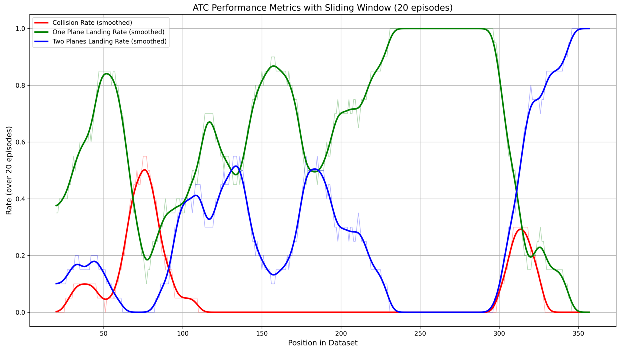

Figure 7: Rewards for the Two Plane Scenario

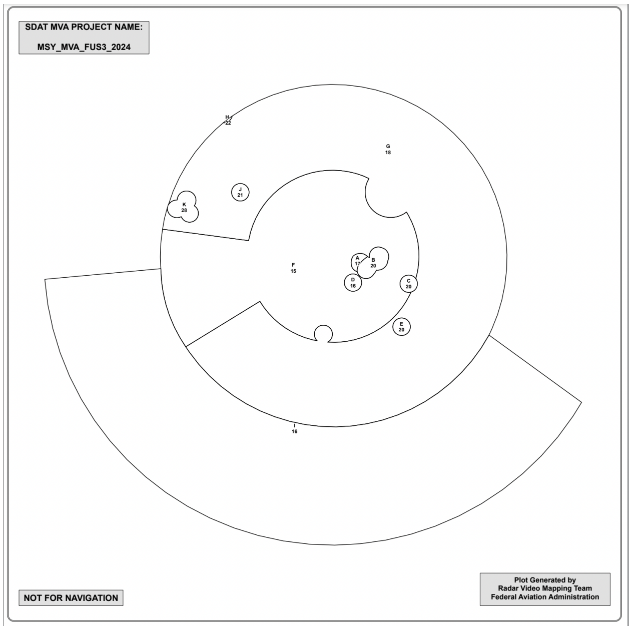

Figure 8: KNEW airport MVA map at 3 nautical miles resolution.