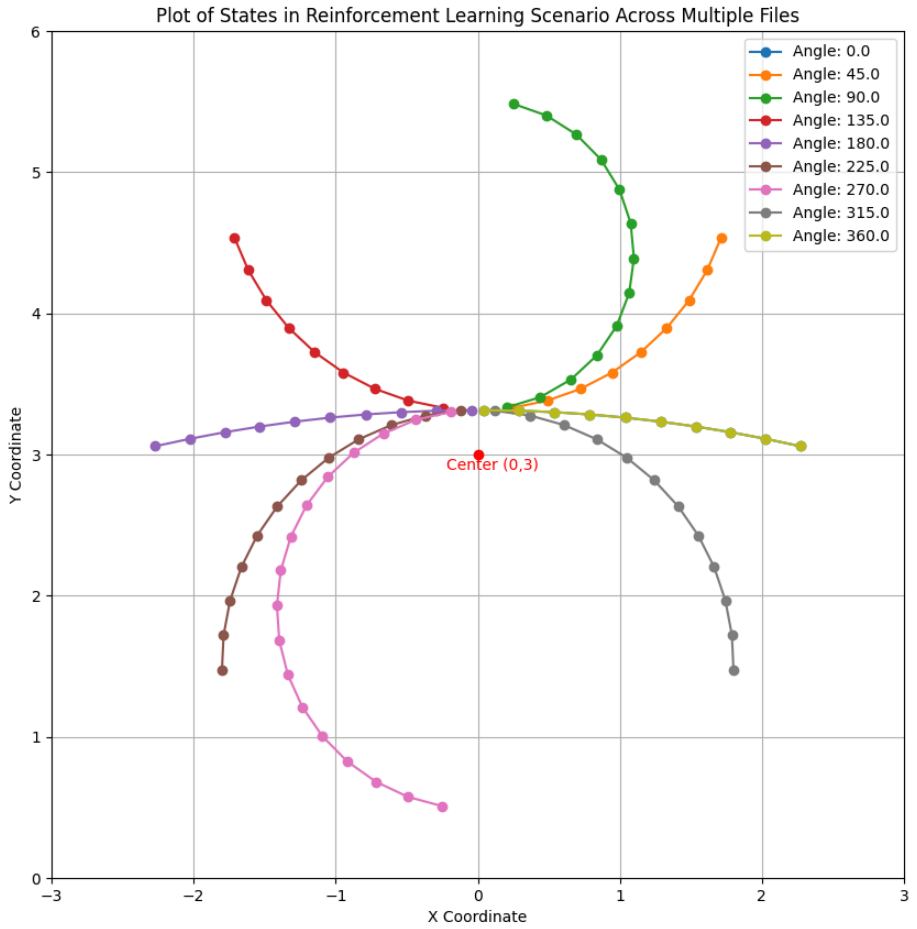
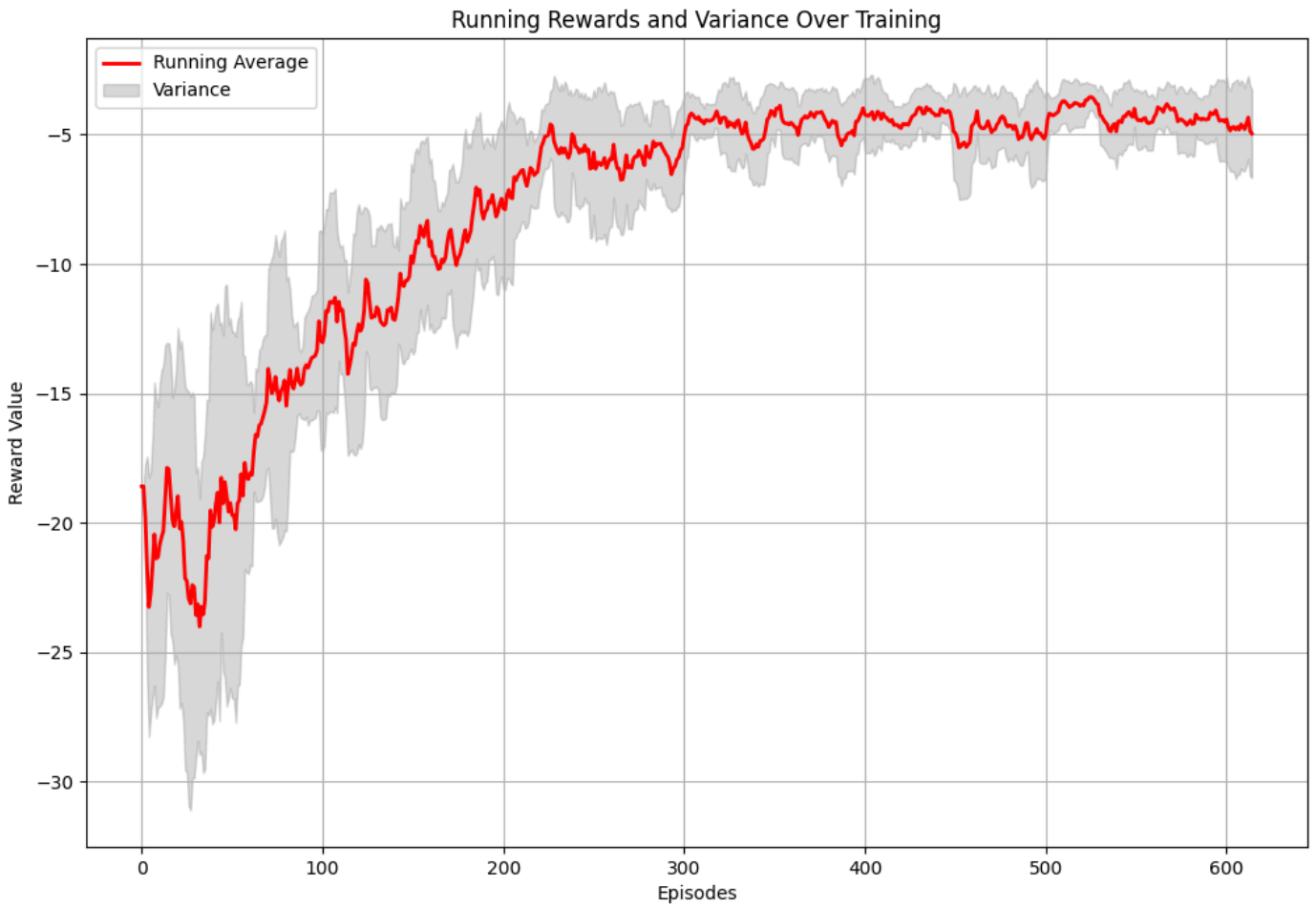
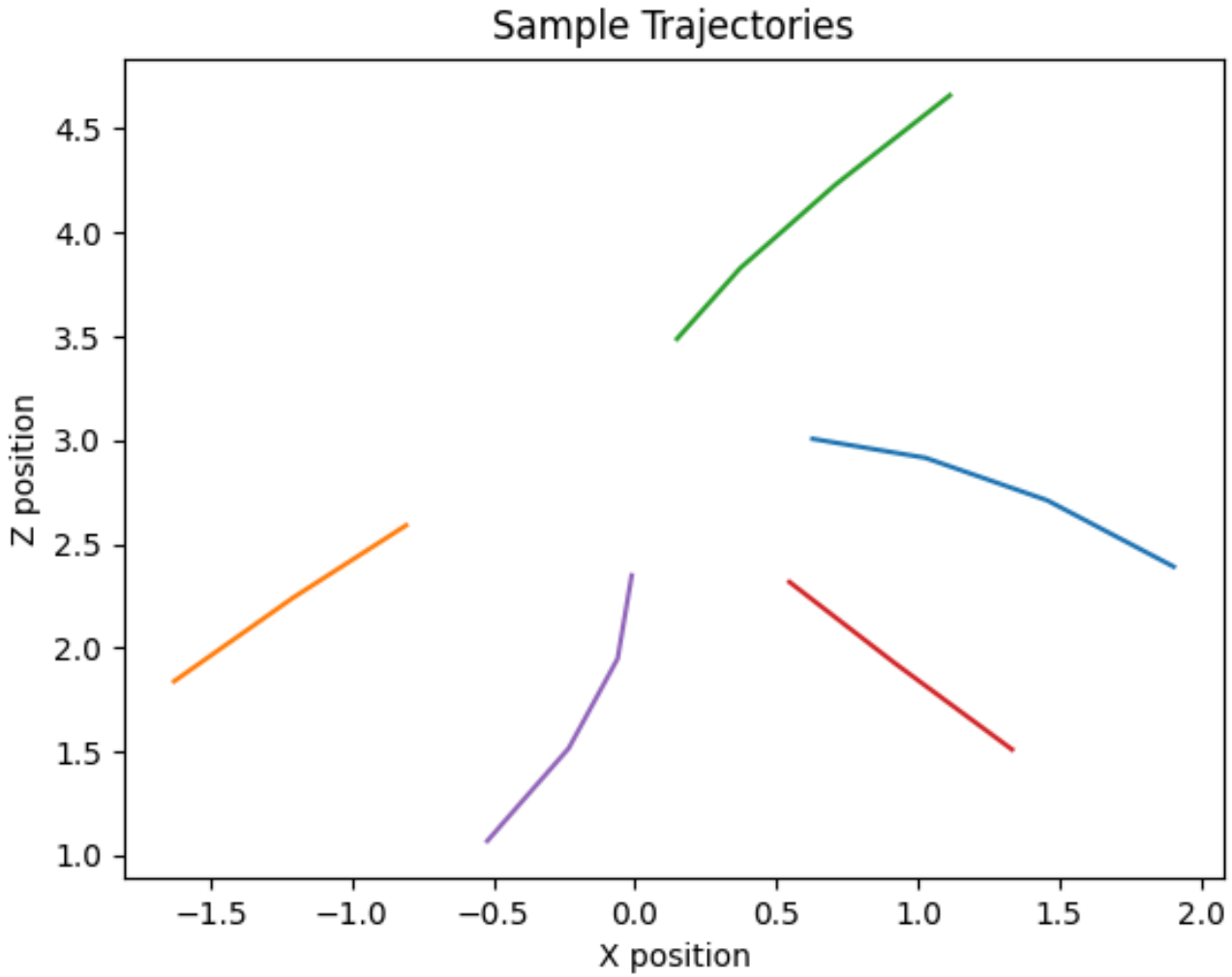
Agile Trajectory Generation for Tensile Perching with Aerial Robots

Progress Update

- Previous Demos
  - Generated a set of demonstrations
  - Converted them match the 2-level waypoint system
- Optimised Demos
  - Generated a set of demonstrations
  - Converted them match the 2-level waypoint system
- Reward = - distance to branch
- Trained Base Algorithm (without demonstrations)
  - Show this
- Adapted


Plot of States in Reinforcement Learning Scenario Across Multiple Files
Angle: 0.0
Angle: 45.0
Angle: 90.0
Angle: 135.0
Angle: 180.0
Angle: 225.0
Angle: 270.0
Angle: 315.0
Angle: 360.0
Center (0,3)
Y Coordinate
X Coordinate
6
5
4
3
2
1
0
−3
−2
−1
0
1
2
3


Running Rewards and Variance Over Training
Running Average
Variance
Reward Value
−5
−10
−15
−20
−25
−30
0
100
200
300
400
500
600
Episodes


Sample Trajectories
Z position
X position
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

From previous update

- Run on current data
  - Demonstrations
    - Hann
      - Generate a set of optimised trajectories from Hann's code. - Done
- Learning from Demonstrations
  - Currently priming the replay buffer with the demonstration data.
  - True algorithm maintains two different buffers and produces different updates. - Done - Cancelled training as I want to fix issues from baseline system.
- Two Level System
  - TODO
  - Using curves for waypoints.
  - Tuning the maximum distance between waypoints.

Next Steps

- Issues faced
  - Resetting
  - Reward Function
    - done based on hitting tether to branch.
    - collision - penalise contact between drone and branch.
    - Trajectory Smoothness Term
- Learning from Demonstrations System
- By next week: Approaching stage finished with comparison results between SAC and SACfD.