AERO 489
Project #2: Final
5/5/2025
Ian Wilhite

1) I chose to implement the 2-rope ball environment.
    a. The environment includes a random starting horizontal velocity, bar angle, angular frequency, and vertical velocity.
    b. The goal as stated was to raise the ball to a height of 10 without falling off the bar, and terminal conditions to ensure the model was meeting real-world constraints.
2) The reward structure and agent terminal conditions were structured to aide the agent to complete the assigned task and penalize suboptimalities in its completion of the task.
   Reward structure:
    a. Proportional height incentive – rewarding the agent for positive movement towards the goal condition.
    b. Proportional time disincentive – penalizing the agent proportionally for longer solutions.
    c. Nonlinear ball position penalty – minor penalty for small deviations from the middle of the ball, and higher penalties for allowing the ball to be closer to the edges of the bar.
    d. Minor maximum angle penalty – disincentivizing the agent from tilting the bar a large amount because of the time required to untilt the bar.

   Terminal condition structure:

    a. Checking the x position of the bar is not off the bar – if the ball has fallen off the edge of the bar the trial has ended.
    b. Checking the bar is upright – ending the trial if theta between negative and positive one-half pi.
    c. Checking the height is greater than negative 10 – this was added to prevent agent from falling into the void during training.
    d. Checking the height is greater than 10 – the success condition.

3) I chose to implement PPO because of its broad convergence in a variety of environments, and its notorious ease to train. During the training process, I varied

the ratio between the time and height proportional rewards, as well as the various minor penalties provided until the agent satisfactorily performed.

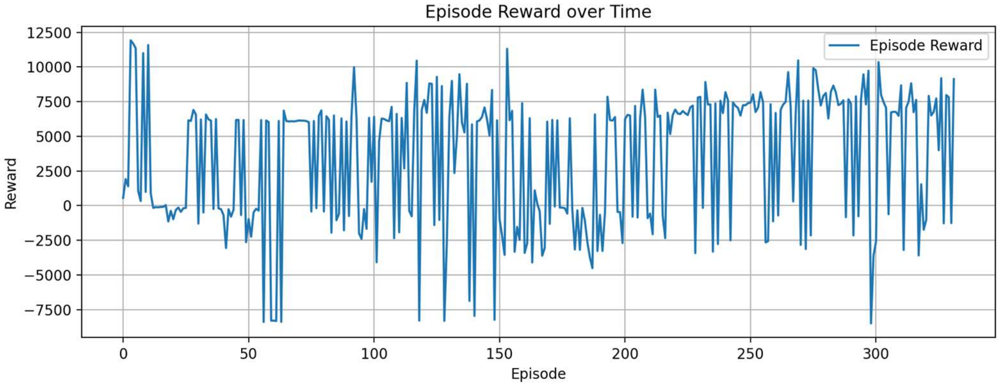
4) Short report:

This project allowed me to learn how to render real world scenarios and to apply the concepts of RL to real situations that I may face in industry. This project truly served to combine many of the skills of this course into a single relevant project relevant to the ways that these tools are currently being applied.

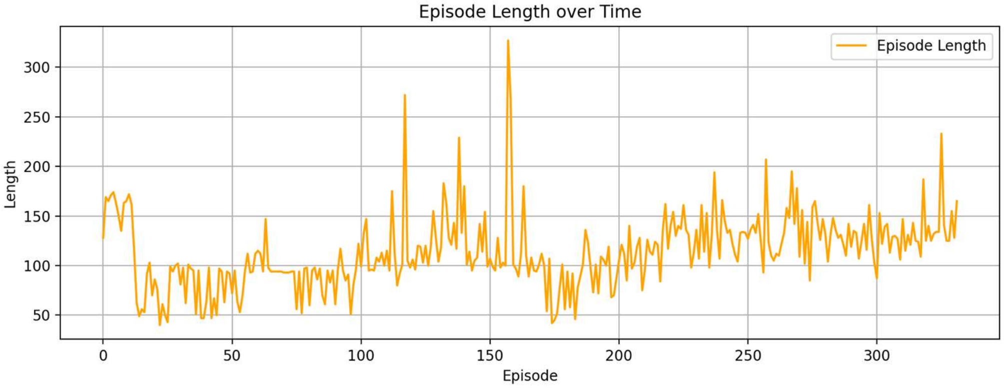
In designing the environment, a major aspect was determining which variables made a major impact on the state and would be required to fully describe any possible given state. Many variables excluded to decrease training time, and therefore the minimum number of variables possible were provided. The horizontal position of the bar and angle were added to represent the position of the system. Their derivatives were added to allow the agent to predict the motion of the system into the future. Their second derivatives were not added because the action the agent took would directly affect it, and therefore the agent did not need to know its own previous action, as it only needed to know what to do next. The height of the bar was intentionally not added to prevent the agent from learning to jerk the cables near the beginning or end, but rather to encourage the agent to act consistently throughout the trial.

The reward structure was revised continuously while the agent was being trained. The combination of linear, nonlinear, and weightings of rewards greatly changed the reaction of the agent to various states, and therefore how it responded to it. Reward shaping drastically affected nearly every aspect of the model's behavior and performance and eventually led to the success of the model. After pivoting to SAV, in tuning, I found that I needed to lower the learning rate to secure a more stable convergence, which finally secured a success rate of 79% in the randomly initialized environment.

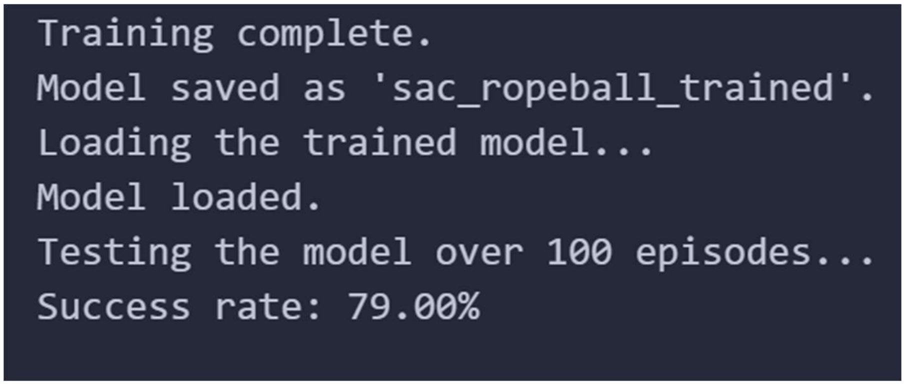
I chose to implement Proximal Policy Optimization (PPO) initially because of its known simplicity in convergence as I was testing the environment, however, I chose to pivot to Soft Actor Critic (SAC) for its ability to perform well in environments with high entropy, making it more suitable for modeling a system of multiple variables and lower training times.



The reward over episode graph shows the improvement of the model, both initially and in stability as the frequency of success increases but could still allow for another decrease in the learning rate and additional trials to allow the model to succeed with higher frequency.

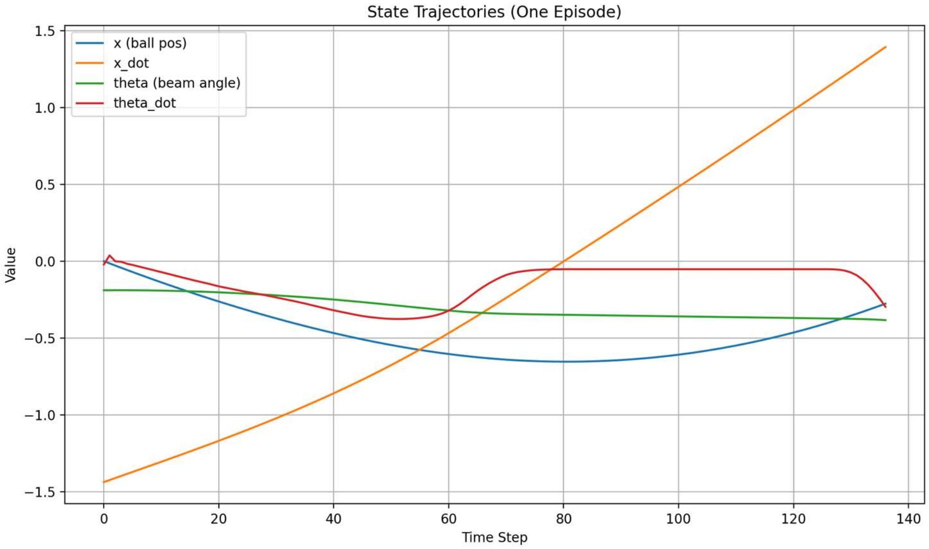


The episode length over time graph shows another perspective of the convergence strength. Initially, there is a sharp decrease in episode time as the model identifies the time penalty, and starts working around it as it learns to stabilize the ball. Around episode 200, the episode length started to rise alongside consistent episodic success, where the model was learning what tradeoffs were worth losing the time penalty points.

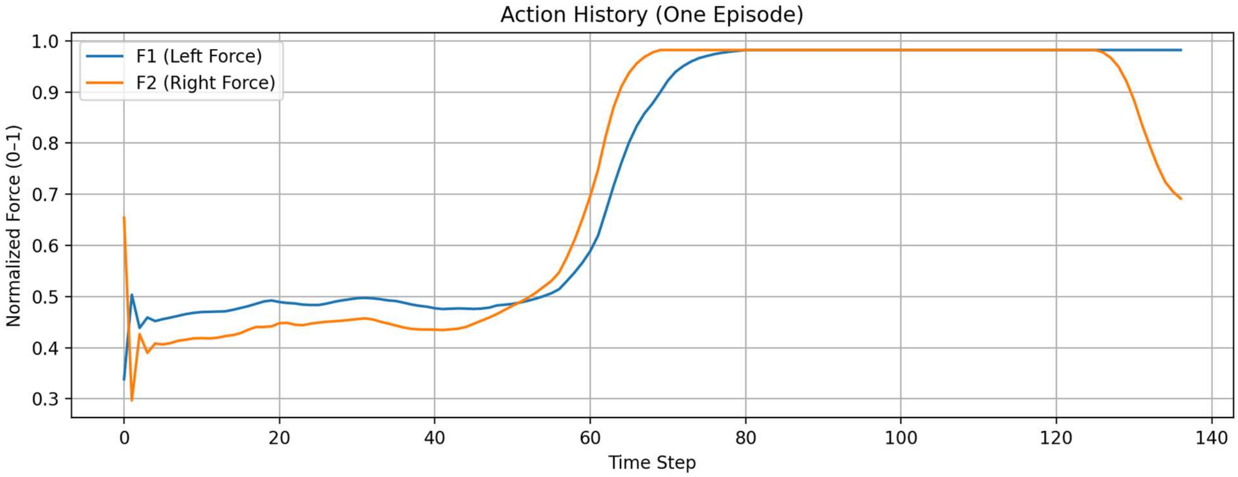


```
Training complete.
Model saved as 'sac_ropeball_trained'.
Loading the trained model...
Model loaded.
Testing the model over 100 episodes...
Success rate: 79.00%
```

To evaluate the model, 100 episodes with random initial angles, angular velocities, and ball speeds were performed, and the success rate for the trials were identified. The agent succeeded in 79 of the 100 episodes, for a success rate of 79% and securely meeting the established threshold of 70%.



The state trajectories plot indicates the agent is able to observe the negative ball position and velocity, and apply a positive torque to counteract the balls motion. The agent is able to correct for and maintain the bars angle, and minimizes the bar's angular acceleration then maintains it.



The system can be observed balancing the ball initially until time step 60, then maximizing a consistent force to raise the bar to the desired height as quickly as possible. The agent can be observed near the end of the trial as accounting for the ball's position crossing the center of the bar and beginning to tilt the bar to correct for overshoot.

Overall, the agent has learned to stabilize and raise the bar to the desired position with satisfactory success. The modeling of the environment represents skills used in industry, while the learning and tuning process is a final application of the skills covered in the second half of the course. This has been certainly one of the most unique classes I have taken, and I truly believe that I have learned a lot from it.