Assignment: Implementing Adaptive Cruise Control using Reinforcement Learning

Instructor: Yaser Fallah March 2025

Overview

A working codebase for a speed-following task using Reinforcement Learning (RL) has been provided to you. In this assignment, you will extend this implementation to develop an Adaptive Cruise Control (ACC) system. Your ACC agent should not only track a reference speed profile but also maintain a safe following distance from a lead vehicle. The target distance range should be between 5 to 30 meters.

Additionally, the acceleration and deceleration values must be constrained within near-realistic physical limits to ensure a feasible and practical simulation of ACC behavior.

Assignment Objectives

Your task is to enhance the provided code by:

1. **Model and Hyperparameter Adjustments:**
   - Experiment with different RL models (e.g., SAC, PPO, TD3, DDPG).
   - Adjust hyperparameters such as learning rate, batch size, buffer size, τ , γ, and the entropy coefficient.
2. **Implementation of Adaptive Cruise Control:**
   - Modify the environment to include a lead vehicle whose position changes dynamically.
   - Ensure the ego vehicle (controlled by the RL agent) maintains a safe distance from the lead vehicle within the range of 5 to 30 meters.
   - Modify the reward function to penalize being too close (less than 5m) or too far (more than 30m) from the lead vehicle.
   - Apply realistic acceleration and deceleration constraints to ensure smooth and practical vehicle behavior.
3. **Episode Length and Reward Structure Modifications:**

- Experiment with different episode lengths (chunk sizes) and evaluate their effect on performance.
- Design and implement a reward function that encourages smooth speed adjustments while maintaining a safe following distance.

4. **Evaluation and Reporting:**
   - Test the modified model on the entire 1200-step dataset in one go.
   - Generate plots comparing the ego vehicle's speed and distance to the reference speed and desired following distance.
   - Include additional plots such as:
     – Jerk (rate of change of acceleration) to assess ride comfort.
     – Speed difference (ego vehicle speed vs. lead vehicle speed).
     – Distance over time to ensure the following distance remains within the specified range.
   - Compute and report relevant statistics, such as:
     – Mean and variance of jerk to measure comfort.
     – Mean absolute speed error.
   - Write a comprehensive report (in PDF format) that includes:
     (a) A description of the modifications made to the model, hyperparameters, environment, and reward structure.
     (b) A discussion of how each modification affects the agent's performance.
     (c) Visualizations (plots) of the test results, including jerk, speed difference, and distance.
     (d) Reported statistics to evaluate how well the ACC is implemented.
     (e) Your interpretation and understanding of the impact of these changes.

Systematic Implementation Guide

To ensure a structured approach to implementing the ACC system, follow these steps:

1. Understanding the provided code: Start by analyzing the existing speed-following RL implementation. Identify key components such as the state space, action space, and reward function.

2. **Modify the environment:**
   - Introduce a lead vehicle with a dynamic position.
   - Define a safe following distance range (5 to 30 meters).

3. **Update the reward function:**

- Penalize unsafe following distances.
- Encourage smooth transitions with minimal jerk.
- Reward maintaining speed similarity with the lead vehicle while keeping a safe distance.

4. **Set realistic constraints:**
    - Limit acceleration and deceleration to feasible values (e.g., within the range of -2 m/s2 to 2 m/s2).
    - Ensure smooth transitions between acceleration states.
5. **Train the RL agent:**
    - Experiment with different RL models (SAC, PPO, TD3, DDPG).
    - Adjust hyperparameters to optimize learning.
6. **Evaluate performance:**
    - Test the model over the full dataset.
    - Generate and analyze plots for speed, distance, jerk, and speed difference.
    - Compute statistical metrics to assess performance.
7. **Write the report:**
    - Summarize modifications and results.
    - Include visualizations and statistical insights.
    - Provide a discussion on model performance and improvement areas.

Submission Requirements

- Submit the modified code (well-commented) in a compressed file (e.g., .zip or .tar).
- Submit a PDF report containing your analysis, experimental results, and discussions.

Evaluation Criteria

- Code Quality: Clarity, documentation, and correctness of modifications.
- Adaptive Cruise Control Implementation: Proper handling of distance constraints and speed adjustments.
- Experimental Analysis: Depth of exploration regarding different models, hyperparameters, episode lengths, and reward structures.

- Report Quality: Clarity, organization, quality of visualizations, and thoroughness of discussion.
- Understanding: Your ability to explain how each change affects the performance of the ACC agent.
- Statistical Evaluation: Quality and completeness of reported statistics to assess the effectiveness of ACC.

Deadline

Please submit your assignment by the deadline (see webcourses).

Additional Notes

- Ensure that your report is self-contained and clearly explains your methodology and findings.
- Collaboration in discussion is allowed, but the final submission must be your original work.
- If you use additional references, please cite them appropriately in your report.