Space Cargo Delivery with RL

Why:

The precision landing of spacecraft has advanced significantly, exemplified by SpaceX's success with the Falcon 9 rocket in 2015, showcasing advanced control algorithms and optimization.
Inspired by this, our project applies reinforcement learning (RL) to cargo delivery services on other planets. In the future, when Mars is colonized, the use of thruster-equipped cargo delivery may become commonplace. These cargos will need to explore various terrains and find a safe landing space. To achieve this, the cargo must determine the optimal thrust and direction to ensure a stable landing while minimizing fuel consumption and avoiding hazards like craters and large rocks. Our proposal focuses on developing a path planning and control mechanism using RL techniques to enhance landing precision and efficiency.

Conventional Algorithms:

The cargo equipped with thrusters works similarly to lunar landers. Conventional algorithms for the lunar lander problem, such as PID and bang-bang control, are easy to implement but are limited in handling complex dynamics, leading to imprecise outcomes. More advanced methods like LQR, optimal control, and Model Predictive Control (MPC) offer improved performance by managing complex constraints but require accurate models and can be computationally intensive, reducing adaptability. Additionally, trajectory optimization lacks flexibility, and bang-bang control may introduce oscillations. In contrast, RL provides a more robust solution by learning adaptive policies from environmental interactions, effectively handling nonlinearities and uncertainties without needing detailed system models, especially in a more complex 3D landing scenario.

Problem statement:

The cargo with thruster must navigate from its current position in 3D space to a designated landing pad at (0,0,0), minimizing fuel usage and avoiding crashes. The cargo operates in continuous 3D space, where its position and velocity are dictated by physics-based dynamics.

The cargo can observe its state, which includes x, y, and z coordinates, velocities in x, y, and z, angular positions (roll, pitch, yaw), angular velocities, and leg contact status. The agent can control the throttle of the main engine, lateral boosters, and orientation engines through discrete actions. The descent is influenced by external factors like gravity and turbulence, and penalties are imposed for excessive fuel usage and unsafe movements. The task ends with a successful landing, a crash, or moving out of the viewport, with performance evaluated based on landing safety and fuel efficiency.

RL cast:

State Space

The state is now a 13-dimensional vector: the coordinates of the cargo in x, y, and z, its linear velocities in x, y, and z, its angular orientation (roll, pitch, yaw), their angular velocities, and three booleans representing whether each leg is in contact with the ground or not.

Action Space

- Do nothing: No engines are fired, allowing the Cargo to free fall.
- Fire left orientation 2 engines at bottom: Rolling the Cargo clockwise.
- Fire right orientation 2 engines at bottom: Rolling the Cargo counterclockwise.
- Fire front orientation 2 engines at bottom: Pitching the Cargo upwards.
- Fire rear orientation 2 engines at bottom: Pitching the Cargo downwards.
- Fire all 4 bottom engines: Slow descent or propel the Cargo upward.
- Fire left lateral engine: Cargo yaw to the right.
- Fire right lateral engine: Cargo yaw to the left.

Reward Structure

- Positive reward for getting closer to the landing pad: Encourages the cargo to navigate toward the target.
- Penalty for moving away from the pad: Discourages deviation from the desired path.
- Penalty for high-speed movement or excessive tilt: Promotes controlled, slow descent and upright positioning to prevent crashes.
- +10 points for each leg that makes contact with the ground: Rewards stable landings with three legs touching down gently.
- -0.03 points per frame for firing any engines: Minimizes overuse of engines, encouraging fuel efficiency.
- -0.3 points per frame for firing all 4 bottom engines: Penalizes excessive vertical thrust, guiding the cargo to use fuel judiciously.
- +100 points for a safe landing; -100 points for a crash: Reinforces the ultimate goal of a successful landing while strongly discouraging crashes.

RL Algorithms and Envisioned Results:

- We will start by trying the Deep Q-Network (DQN) algorithm, as it performs well in discrete action spaces and efficiently learns value functions that map state-action pairs to rewards. However, for comparison, we will also explore state-of-the-art algorithms like Advantage Actor-Critic (A2C) and PPO. These algorithms are better suited for environments with continuous observations and offer greater adaptability and stability when dealing with complex dynamics and changing conditions.
- The performance of these algorithms will be compared in terms of landing success rate, fuel efficiency, crash occurrences, and policy convergence speed.
- Tests will be conducted under varying conditions such as different levels of turbulence and gravity, providing insights into the strengths and weaknesses of each algorithm for real-time control tasks in space exploration.